ABSTRACT

Title of dissertation: LEVERAGING STRUCTURE IN ACTIVITY RECOGNITION: CONTEXT AND SPATIOTEMPORAL DYNAMICS

Sameh Khamis, Doctor of Philosophy, 2015

Dissertation directed by: Larry S. Davis
Department of Computer Science

Activity recognition is one of the fundamental problems of computer vision. An activity recognition system aims to identify the actions of humans from an image or a video. This problem has been historically approached in isolation, and typically as part of a multi-stage system, where tracking for instance is another part. However, recent work sheds light on how activity recognition is in fact entangled with other fundamental problems in the field. Tracking is one such instance, where the identity of each person is maintained across a video sequence. Scene classification is another example, where scene properties are identified from image data. Affordance reasoning is yet another, where the objects in the scene are assigned labels representing what types of actions can be performed upon them.

In this thesis we build a joint formulation for activity recognition, modeling the aforementioned coupled problems as latent variables. Optimizing the objective function for this formulation allows us to recover a more accurate solution to activity recognition and simultaneously solutions to problems like tracking or scene

classification. We first introduce a model that jointly solves tracking and activity recognition from videos. Instead of establishing tracks in a preprocessing step, the model solves a joint optimization problem, recovering actions and identities for every person in a video sequence. We then extend this model to include frame-level cues, where activity labels assigned to people in the same scene are inter-compatible through a scene-level label.

In the second half of the thesis we look at an alternative formulation of the same problem, based on probabilistic logic. This new model leverages the same cues, temporal and spatial, through soft logic rules. This joint formulation can be efficiently solved, recovering both action labels and tracks. We finally introduce another model that reformulates action recognition in the multi-label setting, where each person can be performing more than one action at the same time. In this setting, a joint formulation can solve for all the likely actions of a person through explicit modeling of action label correlations.

Finally, we conclude with a discussion of several challenges and how they can motivate viable future extensions.

LEVERAGING STRUCTURE IN ACTIVITY RECOGNITION: CONTEXT AND SPATIOTEMPORAL DYNAMICS

by

Sameh Khamis

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Larry S. Davis, Advisor
Professor David W. Jacobs
Professor Yiannis Aloimonos
Associate Professor Jimmy Lin, Dean's Representative
Assistant Professor Hector Corrada Bravo

Dedication

To my late mother and grandmother (-2014),

the two wonderful, selfless women who raised me.

Acknowledgments

First and foremost, I would like to thank my advisor Prof. Larry S. Davis for many years of support and guidance. Larry is a remarkably meticulous and caring mentor. Working in his lab, I was provided with all the facilities and support to do research, while still allowed enough independence to pursue topics of interest to me. I definitely could not ask of a more approachable mentor. Despite the relatively large size of the group, he always had and has time for everyone. Larry has been an extremely positive influence on my life in many ways and my time with him left me prepared for a career in research. I am lucky to have had the fortune of working with him.

I would also like to thank my committee members for taking the time to read my thesis. I am especially thankful to Prof. David Jacobs for the insightful comments on my research during my proposal, and Prof. Yiannis Aloimonos for the numerous perceptive big-picture conversations we had.

I was fortunate to work with amazing mentors over the years. I am thankful for the opportunity to work with Christoph Lampert at IST Austria. Christoph is not only a remarkable researcher but is also a humble human being. I have learned a lot from his work, which straddles both theory and practice. I will miss the brainstorming, the foosball games, and most of all the humor.

I am also thankful for my time at Microsoft Research in Cambridge. The support I received working on a completely new problem was unparalleled. I had the pleasure of working with amazing researchers, and I had the privilege to experience

first-hand world-class collaborative research. I will always be grateful to Jonathan Taylor, Jamie Shotton, Cem Keskin, Andrew Fitzgibbon, and Shahram Izadi. I grew as a researcher in your group. Jonathan, I am especially thankful for the collaboration and pair programming, the conversations, and the friendship.

Thanks to all Friends, inside and outside the vision lab. I am thankful for all the amazing people who grew close to me over the last few years. Thanks to my good friends Ioana Bercea, Ben London, and Theodoros Rekatsinas for all the fun nights out. Thanks to Jonghyun Choi, Mohammad Rastegari, and Abhishek Sharma for the heartfelt conversations about grad life. Thanks to Vlad Morariu for the generous help and for being a constant voice of reason. A very special thanks to my (practically) lifelong friend Hossam Isack for always being there for me.

Above all, I owe the utmost gratitude to my family. To my parents who planted the seed for my interest in computer science at a very young age and then encouraged me to pursue graduate degrees. To my late mother and grandmother, your endless support made me who I am today. To my brother Ashraf, thank you for being a real and true friend my entire life, and for lending me your ear whenever I needed to talk. Last but certainly not least, my wife Jessica, thank you for the friendship and the patience, thank you for being the anchor to my sanity throughout the years, and thank you for sharing your wonderful family with me.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Activity recognition is a fundamental problem in computer vision. The main aim of an activity recognition system is to identify the actions of one or more persons from an image or a video. In many settings this problem is essentially intertwined with other fundamental problems in the field. Tracking is one such instance, where the identity of each person is maintained across a video sequence. Scene classification is another example, where scene properties are identified from image data. Affordance reasoning is yet another, where the objects in the scene are assigned labels representing what types of actions can be performed upon them.

Consider the scene in Figure 1.1. The goal of an activity recognition system is to label every person in a video of frames like this one. Attempting this task by analyzing the bounding boxes of the people in the scene discards useful information. Not only is the action a person is performing at a specific frame a cue for her action in future frames, but it is also a cue for the actions of people in her vicinity in the current frame. A person who is waiting to cross the street will likely cross the street in a succeeding frame, and people in her vicinity will likely be also crossing the street. Knowing what the scene represents, whether it is a street scene, a school

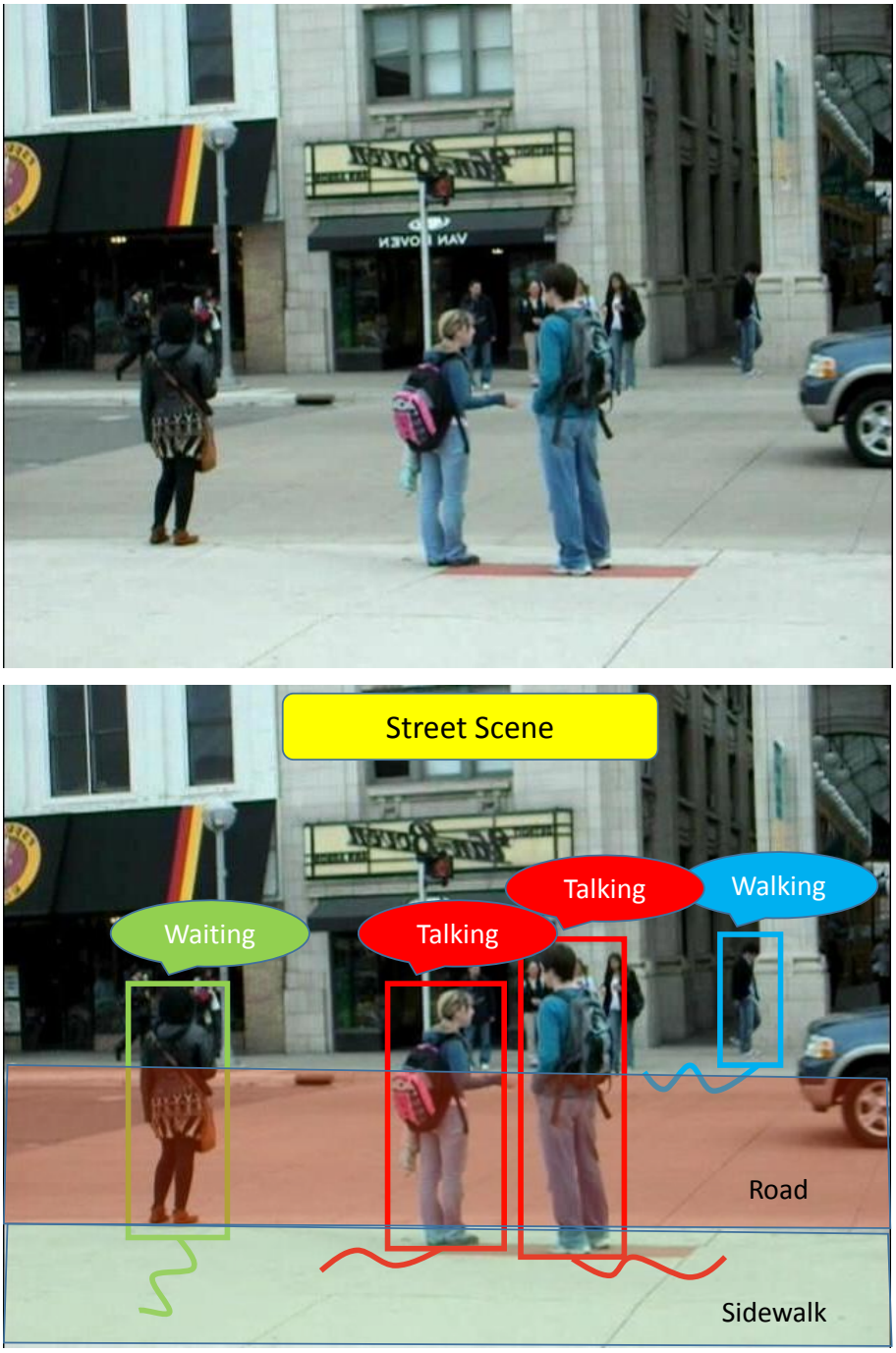

Figure 1.1: Leveraging structure in activity recognition. An action recognition model that integrates all the available spatiotemporal and contextual cues will likely outperform one that does not.

yard, a hallway, or a studio, is another strong cue for what each person might be doing. Additionally, the location of the person in the scene is another cue, *e.g.* people tend to cross roads on crosswalks.

This thesis tackles this problem from a modeling perspective, integrating all these cues into a joint formulation. Activity recognition from realistic videos can benefit from leveraging the structure of the scene. This structure is presented through temporal constraints, where tracking can be utilized, and it can also be presented through spatial constraints, where the actions of multiple people in the same frame are recognized jointly.

1.2 Background

Action recognition research has recently made tremendous strides. In the past few years research has gone beyond the classic single person short video [4, 5] to model action parts, context, object interactions, group activities, and spatio-temporal connections between actors [6–8]. Motivated by the rich spatiotemporal structure of human activity, researchers have explicitly modeled interactions among actions under observation, jointly solving multiple previously independent vision problems. Such interactions include those between scenes and actions (e.g., *road* and *driving*) [9], objects and actions [8, 10] (e.g., *spray bottle* and *spraying*, *tennis raquet* and *swinging*) or actions performed by two or more people [2, 3, 11, 12] (e.g., two people standing versus two people queueing). More complex high level interactions have also been modeled, e.g., by dynamic Bayesian networks (DBNs) [13], CASE

natural language representations [14], Context-Free Grammars (CFGs) [7], AND-OR graphs [6], and probabilistic first-order logic [15, 16].

This work presented in this thesis is closely related to previous work on modeling collective behavior [2, 3, 12]. Choi *et al.* [2] initially introduced this problem, proposing a spatio-temporal local (STL) descriptor that relies on an initial 2.5D tracking step which is used to construct histograms of poses (facing left, right, forward, or backward) at binned locations around an anchor person. These descriptors are aggregated over time, used as features for a linear SVM classifier with a pyramid-like kernel, and combined with velocity-based features to infer the activity of each person. Collective activity is modeled through the construction of the STL feature. In later work, Choi *et al.* [3] extend the STL descriptor by using random forests to bin the attribute space and spatio-temporal volume adaptively, in order to better discriminate between collective activities. An MRF applied over the random forest output regularizes collective activities in both time and space.

Lan *et al.* [12] propose a slightly modified descriptor, the Action Context (AC) descriptor, which, unlike the STL descriptor, encodes the actions instead of the poses of people at nearby locations. The AC descriptor stores for each region around a person a k-dimensional response vector obtained from the output of k action classifiers. Instead of relying on local descriptors alone, Lan *et al.* [11] explicitly model group activity by simultaneously modeling the individual actions, their relation to an overall group activity, and their relation to each other. The structure of the person-person interaction graph is inferred as part of the overall inference task. While this successfully models group activities, it does not directly model the

temporal progression of individual actions or group activities.

1.3 Contributions

This thesis contributes several models to exploit the rich structural cues surrounding human activity recognition in the multi-person setting. These cues are explicitly modeled throughout the thesis and the performance gain is evaluated empirically.

- Chapter 2 introduces a model that jointly solves tracking and activity recognition. Instead of establishing tracks in a preprocessing step, where errors can only propagate forward, the model solves a single objective function, recovering actions and identities for every person in a video sequence.

- Chapter 3 exploits not only the tracking cues but also the frame level cues. The introduced joint model assigns people in the same scene activities that are inter-compatible through a scene-level label. The model employs the formulation from Chapter 2 in the same objective function to add the track-level cues as well.

- Chapter 4 explores a different model based on probabilistic logic. The new model combines the same cues, temporal and spatial, through soft logic rules in a single formulation. Jointly solved, it recovers both actions labels and tracks.

- Chapter 5 looks at a new formulation of action recognition, where each person

can be performing more than one action at the same time. In this setting, a joint formulation can solve for all the likely actions for a person in a given frame by explicitly modeling the correlations of action labels.

Each chapter will introduce the background and related work that is associated with the problem it is solving. Chapter 6 concludes the dissertation and proposes viable future extensions.

Chapter 2: Action Recognition and Identity Maintenance

2.1 Introduction

In this chapter we introduce a novel model for human action recognition from videos. We are motivated by the fact that actions in a video sequence typically follow a natural order. Consider the illustration in Figure 2.1. The person outlined in the left image is queueing, while the person outlined in the right image is waiting to cross. Given the appearance and stance resemblance, a classifier might return similar scores for both actions. However, we can take advantage of their actions at a later time, when the person on the right will be crossing while the person on the left will still be queueing; their actions then become more distinguishable.

One issue that remains with this idea is identity maintenance. A simple approach would be to build the tracks of people detections using appearance models, and then construct an action recognition model that makes use of the identities established from the tracking step. This approach assumes that such tracks are accurate and disregards the advantage of jointly solving both problems under one model. This is most evident with similar appearances and overlapping bounding boxes, where the likelihood of a transition between compatible actions can improve the inference of the identities.

We develop a novel representation of the joint problem. We initially train a linear SVM on the Action Context (AC) descriptor [12], which explicitly accounts for group actions to recognize an individual's action. We use the normalized classifier scores for the action likelihood potentials. We then train an appearance model for identity association. Our association potentials incorporate both appearance cues and action consistency cues. Our problem is then represented by a constrained multi-criteria objective function. Casting this problem in a network flow model allows us to perform inference efficiently and exactly. Finally, we report results that outperform state-of-the-art methods on two group action datasets.

Our contribution in this work is three-fold:

- We propose jointly solving action recognition and identity maintenance under one model.
- We formulate inference as a flow problem and solve it exactly and efficiently.
- Our action recognition performance improves on the state-of-the-art results for two datasets.

The rest of this chapter is structured as follows. In Section 2.2 we survey the related literature and discuss our contribution in its light. We introduce our approach and focus on the problem formulation in Section 2.3. We then discuss the system in details in Section 2.4. We present the datasets in Section 2.5, and report our results quantitatively and qualitatively. And last, we conclude in Section 2.6.

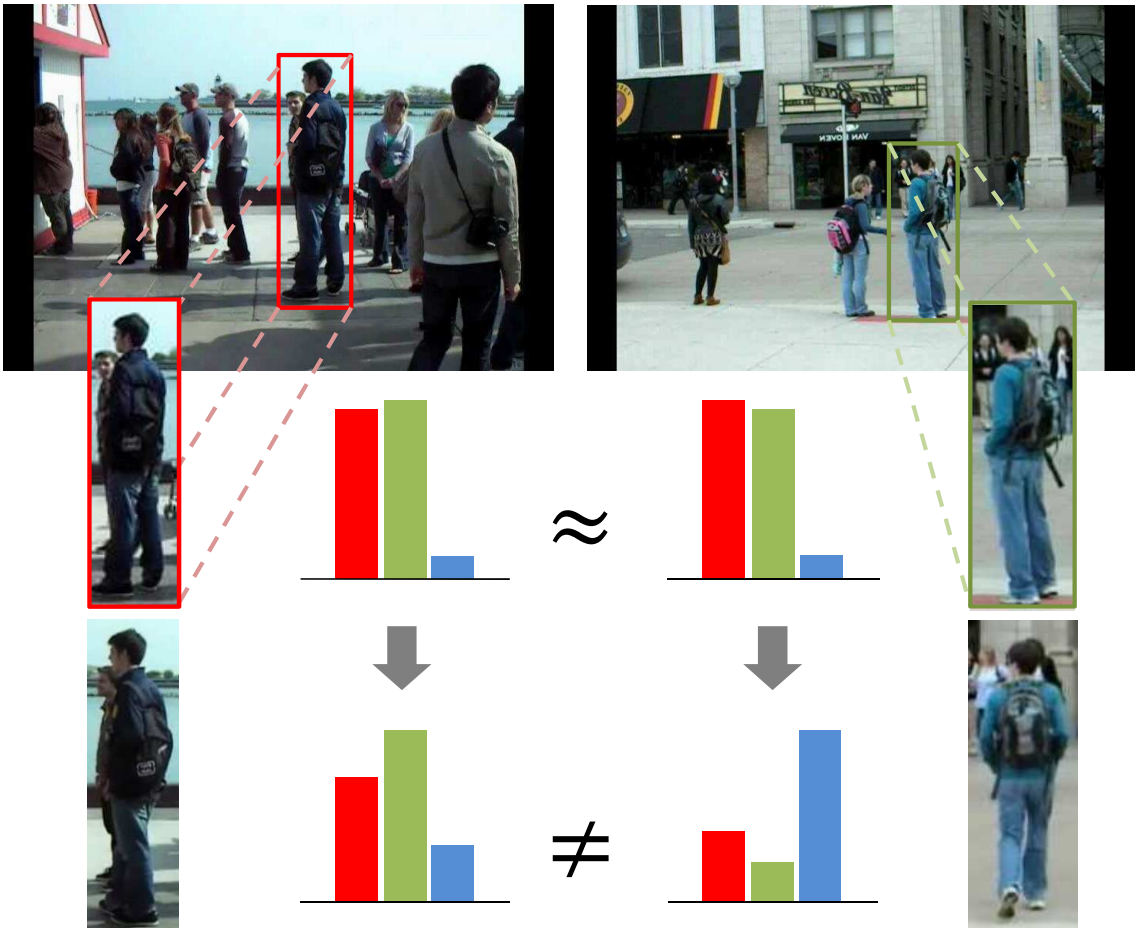

Figure 2.1: How tracking can improve action recognition. Tracking can improve action recognition in the multi-person setting, and we present a model to solve both problems jointly and efficiently.

2.2 Related Work

We survey the action recognition literature and the tremendous breakthroughs that recent research has established in Section 1.2.

To reason about actions over time, most of the surveyed approaches require that people or objects are already detected and tracked [2,3,6,7,10,12,15,16]. These

tracks can be obtained by first detecting people and objects using detectors such as Felzenszwalb *et al.* [17] and then linking the resulting detections to form tracks. For example, the detection based tracking approach of Zhang *et al.* [18] links detections into tracklets using a global data association framework based on network flows. Pirsiavash *et al.* [19] extend this approach while maintaining global-optimality by performing shortest path computations on a flow network. Berclaz *et al.* divide the scene into a network flow problem on a spatio-temporal node grid [20], which they solve using the *k-shortest path* algorithm. This approach, while not requiring the detection of bounding boxes before tracking, results in a significantly larger state-space than [18]. Ben Shitrit *et al.* extend this work by introducing a global appearance model, reducing the number of track switches for overlapping tracks [21].

While performing tracking and activity recognition sequentially simplifies action recognition, since the problem of identity maintenance can be ignored during the recognition step, mistakes performed during the tracking step cannot be overcome during recognition. Motivated by the improved results of explicitly modeling the interactions of multiple vision problems jointly (person-object, person-person, *etc.*), the work presented in this thesis tackles solving both problems, identity maintenance and activity recognition, in a joint formulation.

2.3 Approach

2.3.1 Overview

Our focus in this work is to improve human action recognition. We assume that humans have already been localized, *e.g.*, with a state-of-the-art multi-part model [17], or with background subtraction if the camera is stationary. Our representation for a detected human figure is based on Histogram of Oriented Gradients (HOG) [22], for which we use the popular implementation from Felzenszwalb *et al.* [17]. We then run a two-stage classification process by computing the Action Context (AC) descriptor [12]. We adopt the AC descriptor to model human actions in the context of actions performed by nearby people; however, to reason about these actions over time, we integrate this representation into our joint model, instead of pre-computing track associations. We use this representation to train the action likelihoods in our model. Figure 2.2 illustrates the overall flow of analysis. and the details are presented in Section 2.4

Finally, to reason about appearance across time, we augment our representation with the blurred and subsampled bounding boxes (detections) in *Lab* color space. We use this representation to train the association likelihoods used in our model.

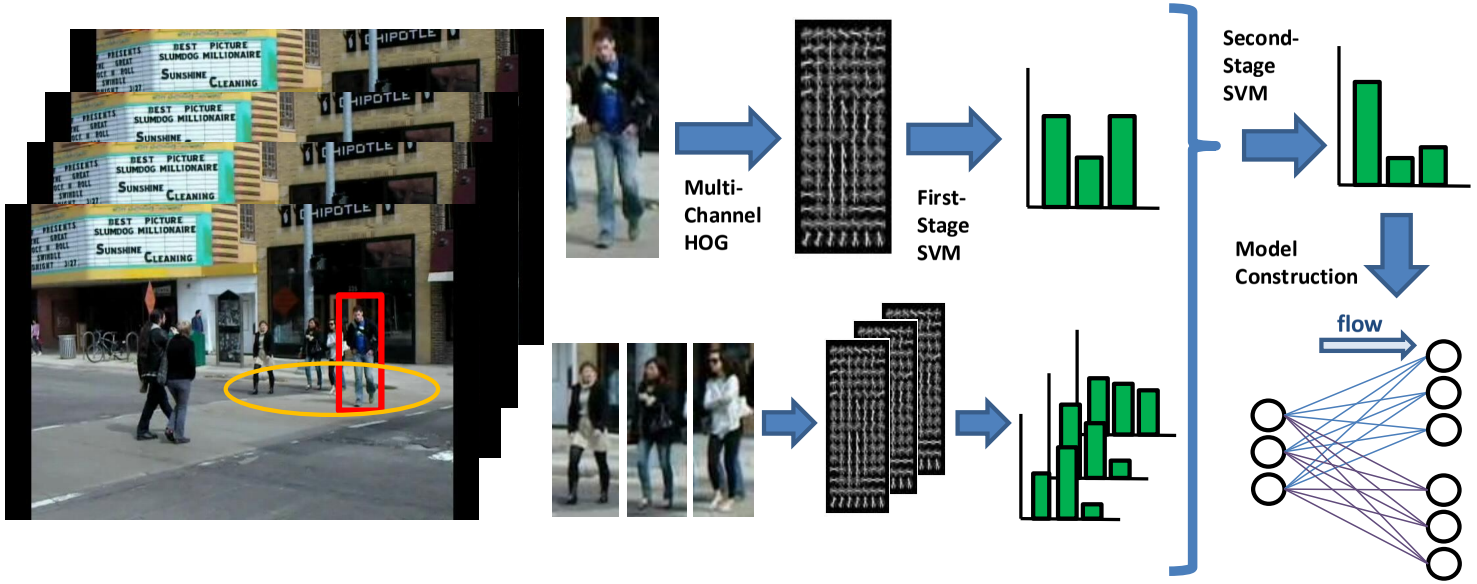

Figure 2.2: An overview of our system. We first extract features to identify actions using a two-stage classification approach. We then integrate those features alongside appearance-consistency features into a joint model. Optimizing the objective function for the model, we recover actions and tracks for people in the video.

2.3.2 Formulation

Similar to [18, 19], given human detections, we pose the problem of identity maintenance as a network flow problem, which allows us to obtain the solution exactly and efficiently, while focusing on our final goal of activity recognition.

We use i, j, and k to denote the indices of human detections in a video sequence, while a, b, and c are used to denote actions. We also define $\mathcal{P}(i)$ to be the set of candidate predecessors for detection i from prior frames, and similarly $\mathcal{S}(i)$ to be the set of candidate successors of detection i from subsequent frames. We indicate the action and the identity of a detected person i by y_i and z_i, respectively. We can then formulate our model as a cost function over actions and identities represented as

$$F(\mathbf{y}, \mathbf{z}) = \sum_i \sum_a \Big[u_a(i) + v'_a(i) \Big] \mathbb{1}(y_i = a), \tag{2.1}$$

where $u_a(i)$ is the classification cost associated with assigning action a to person i, and $v'_a(i)$ is the associated tracking cost. Commonly, $\mathbb{1}(.)$ is defined as the indicator function.

We define the classification cost $u_a(i)$ to be the normalized negative classification score of person i performing action a. The details of the classifier training procedure is in Section 2.4.2.

Since a detection could designate a new person entering the scene, we define our tracking cost as

$$v'_a(i) = \begin{cases} v_{ab}(i,j) & \text{if } \exists j \in \mathcal{P}(i) \text{ s.t. } z_i = z_j,\ y_j = b, \\ \lambda_0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where $v_{ab}(i,j)$ is the transition cost that links "person i performing action a" to a previously tracked "person j performing action b". If the newly detected person i does not sufficiently match any of the people previously tracked, the model incurs a penalty represented by the tuning parameter λ_0, and a new track is established. We define the transition cost $v_{ab}(i,j)$ as

$$v_{ab}(i,j) = \lambda_d\ d(i,j) - \lambda_c \log(p_{ab}), \tag{2.3}$$

which is a mixture of an appearance term and an action consistency term. The appearance term measures the similarity between person i and person j with a distance metric $d(i,j)$, and the action consistency term measures the prior probability p_{ab} of a person performing action a followed by action b. The tuning parameters λ_d and λ_c weigh the importance of those two terms. The models for calculating both the appearance distance metric $d(i,j)$ and the action co-occurrences p_{ab} are provided in Section 2.4.3.

Maximum-a-posteriori (MAP) estimation in our model can be formulated as the minimum of an integer linear program (ILP). We define the following program

$$
\begin{aligned}
\min_{\{\mathbf{e},\mathbf{t},\mathbf{x}\}} \quad & \sum_i \sum_a \Big[(u_a(i) + \lambda_0) e_a(i) + \\
& \qquad \sum_{j \in \mathcal{P}(i)} \sum_b (u_a(i) + v_{ab}(i,j)) t_{ab}(i,j) \Big], \\
s.t. \quad & e_a(i) + \sum_{j \in \mathcal{P}(i)} \sum_b t_{ab}(i,j) = \\
& \qquad x_a(i) + \sum_{k \in \mathcal{S}(i)} \sum_c t_{ca}(k,i) \quad \forall i, a \\
& \sum_a \Big[e_a(i) + \sum_{j \in \mathcal{P}(i)} \sum_b t_{ab}(i,j) \Big] = 1 \quad \forall i \\
& \{\mathbf{e}, \mathbf{t}, \mathbf{x}\} \in \mathbb{B}^n,
\end{aligned}
\tag{2.4}
$$

where variable $e_a(i)$ denotes the entrance of person i into the scene performing action a, while variable $t_{ab}(i,j)$ denotes the transition link of person i performing action a to person j performing action b. Finally, variable $x_a(i)$ denotes person i exiting the scene after performing action a. The entrance, transition, and exit variables are defined to be binary indicators. The costs $u_a(i)$ and $v_{ab}(i,j)$ are as previously defined.

Minimizing the program in Equation 2.4 is equivalent to inference in the model from Equation 2.1. A detected human figure would always encounter a classification cost, whether it is linked to a previously tracked detection, or is entering the scene for the first time. Consequently, it will either incur the transition cost to link it to the previously tracked detection, or incur the penalty of not having a sufficiently matching predecessor. The two constraints enforce a valid assignment according to Equations 2.1 and 2.2.

The variables $\mathbf{e}$, $\mathbf{t}$, and $\mathbf{x}$ always recover a unique assignment for $\mathbf{y}$ and $\mathbf{z}$. Specifically, if detection i just entered the scene, it will be assigned action $y_i = a$ for which $e_a(i) = 1$ and its identity z_i will be assigned to an unused track number. Otherwise, detection i will be instead linked to a previous detection; in that case, it will be assigned action $y_i = a$ for which $t_{ca}(k, i) = 1$ and the identity will propagate from that previous detection: $z_i = z_k$.

The ILP in Equation 2.4 represents a network flow problem. In fact, the first constraint of the ILP is the "flow conservation constraint" (or *Kirchoff's Laws*). However, the second constraint, which we refer to as the "explanation constraint", is not typically encountered in the minimum cost flow problem. In our case, it enforces that an action and an identity be assigned to every person detected in the video.

Figure 2.3 illustrates the flow graph of an example with 3 frames, 5 detections, and 3 possible actions per person. Every grouped subset of nodes represents a detection, and the nodes in the subset are potential actions for that detection. Every detection forms a complete bipartite graph with its predecessors (previous frames) and successors (following frames). Here people in every frame are connected to those in the previous frame, but that can be generalized to any subset of people in any number of frames. The flow goes from the source node to the sink node assigning actions and identities that minimize our integer linear program in Equation 2.4. By enforcing the "explanation constraint", we are guaranteed an action and an identity for every person in the graph. The flow of the minimum cost in the network uniquely assigns actions and identities to every detected person in the video sequence. The

colored arcs in the diagram represent an example of a valid complete assignment (corresponding to a flow of minimum cost) for the frame sequence at the bottom. The person outlined in green enters in the first frame, performs the first action for the entire sequence, and exits in the final frame, while the person outlined in red enters in the second frame, performs the second action, before exiting at the final frame.

2.3.3 Inference

While minimim cost flow problems with side constraints can generally be solved by *Lagrangian Relaxation* (also known as *Dual Decomposition*) [23], the form of our constraints allows us to provide fast alternative solutions. As shown in Equation 2.4 and Figure 2.3, our formulation uses constraints on sets of nodes. We relax the binary constraint in Equation 2.4 to an interval constraint and directly solve the linear program using a fast interior-point solver. To improve the inference speed, we only connect people with overlapping bounding boxes in consecutive frames. Solving the cost function exactly takes an average of 1.2 seconds for an average sequence length of 520 frames, where each sequence is subsampled every ten frames during model construction.

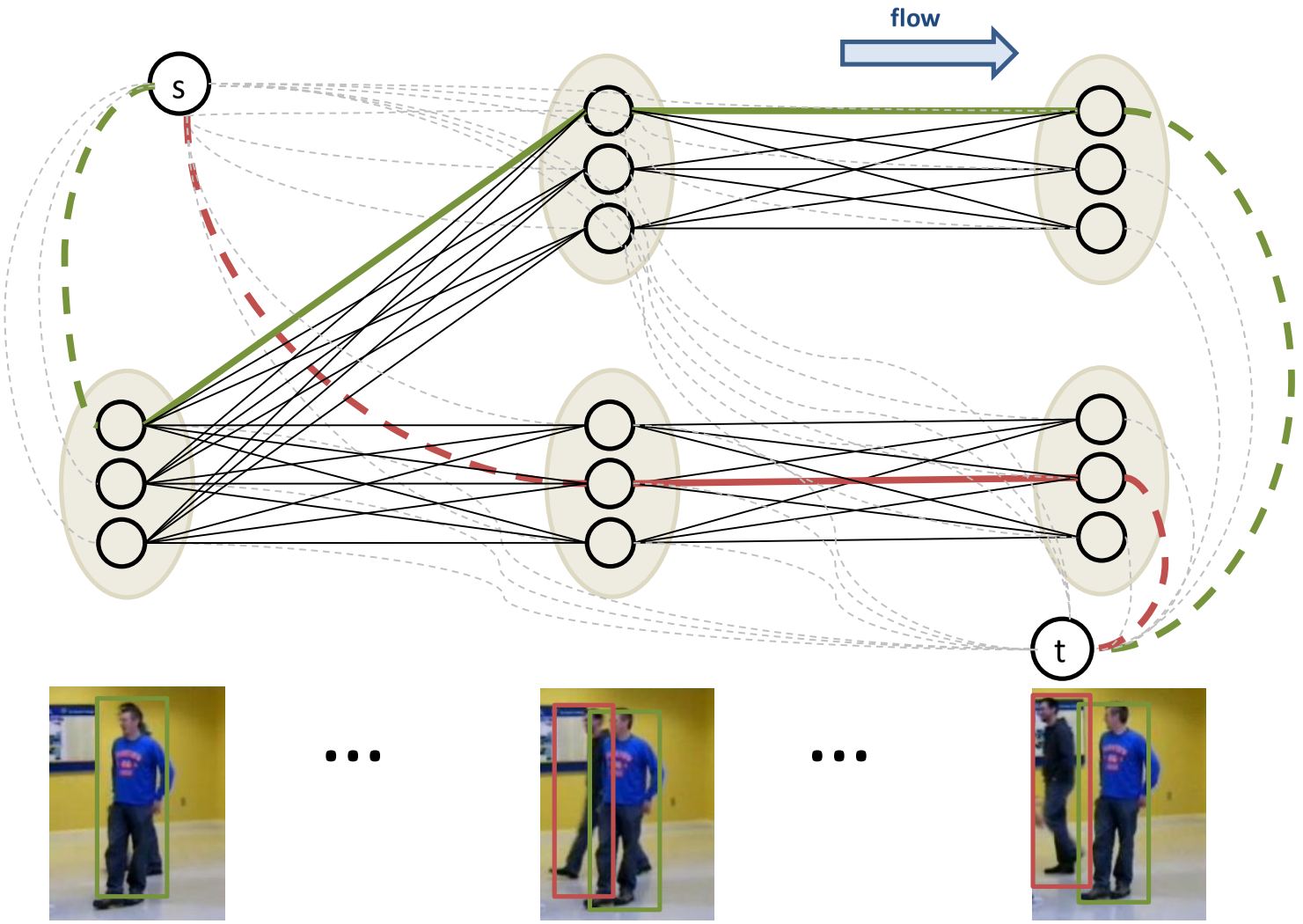

Figure 2.3: An illustration of our flow model. The flow of the minimum cost in the network uniquely assigns actions and identities to every detected person in the video sequence. Section 2.3.2 provides the technical details.

2.4 Learning the Potentials

2.4.1 Piecewise Training

Since inference in our model is exact and latent variables are absent, global training approaches become not only possible, but deterministic. However, for practical reasons, we chose to use piecewise training [24]. Piecewise training involves dividing the model into several components, each of which is trained independently. We are motivated by recent theoretical and practical results. Theoretically speaking, piecewise training minimizes an upper bound on the log partition function of the model, which corresponds to maximizing a lower bound on the exact likelihood. In practice, the experiments of [24, 25] show that piecewise training sometimes outperforms global training, even when joint full inference is used. We choose to divide our model training across potentials, i.e., we train the three groups of potentials–unary action, binary action consistency, and binary appearance consistency–independently from each other. The tuning parameters that weigh the importance of the individual terms were set manually through visual inspection.

2.4.2 Action Potentials

We now describe how we train our action likelihood potentials. We use the AC descriptor from Lan *et al.* [12]. We utilize HOG features as the underlying representation. We then train a multi-class linear SVM using *LibLinear* [26]. Next, a bag-of-words style representation for the action descriptor of each person is built.

Each person is represented by the associated classifier scores, and the strongest classifier response for every action in a set of defined neighborhood regions in their context.

The descriptor of the i-th person becomes the concatenation of their action scores and context scores. The action scores for person i, given A possible actions, become $\mathbf{F}_i = [s_1(i), s_2(i), \ldots, s_A(i)]$, where $s_a(i)$ is the score of classifying person i to action a. The context score, defined over M neighborhood regions, is

$$\mathbf{C}_i = \Big[\max_{j \in \mathcal{N}_1(i)} s_1(j), \ldots, \max_{j \in \mathcal{N}_1(i)} s_A(j), \ldots, \max_{j \in \mathcal{N}_M(i)} s_1(j), \ldots, \max_{j \in \mathcal{N}_M(i)} s_A(j) \Big], \tag{2.5}$$

where $\mathcal{N}_m(i)$ is a list of people in the m-th region in the neighborhood of the i-th person. We use the same "sub-context regions" as [12]. We then run a second-stage classifer on the extracted AC descriptor using the same multi-class linear SVM implementation of *LibLinear* [26]. The classifier scores are negated and then normalized using a softmax function, and finally incorporated as the unary action likelihood potentials $u_a(i)$, which assign action a to person i.

2.4.3 Association Potentials

To track the identities of the targets in our video sequences, we train identity association potentials and incorporate them in our model. Our association potentials use both appearance and action consistency cues. The appearance cues are trained using the subsampled color channels as features. We train for a Mahalanobis distance

matrix M to estimate the similarity between detections across frames. The distance matrix is learned so as to bring detections from the same track closer, and those from different tracks apart [27]. This is formulated as

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \sum_{\mathcal{T}_k} \Bigg[\sum_{i,j \in \mathcal{T}_k} (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j) - \sum_{i' \in \mathcal{T}_k, j' \notin \mathcal{T}_k} (\mathbf{f}_{i'} - \mathbf{f}_{j'})^T \mathbf{M} (\mathbf{f}_{i'} - \mathbf{f}_{j'}) \Bigg], \tag{2.6}$$

where $\mathcal{T}_k$ is the k-th track and $\mathbf{f}_i$ is the feature vector of the i-th person. We solve for M using the fast Large Margin Nearest Neighbor (LMNN) implementation of [28]. The distance between two people i and j can then be defined as

$$d(i,j) = (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j). \tag{2.7}$$

The action consistency cues are estimated using the groundtruth action labels from the training set. We count pairwise co-occurrences of actions on the same track plus a small additive smoothing parameter α. The counts are normalized into the pairwise co-occurrence probabilities p_{ab} of action pairs a and b.

2.5 Experiments

2.5.1 Datasets

We use the group actions dataset from [2] and its augmentation from [3] to evaluate our model. The datasets are appropriate since they have multiple targets

in a natural setting, while most action datasets, like KTH [5] or Weizmann [4], have a single person performing a specific action. The original dataset includes 5 action classes: *crossing*, *standing*, *queueing*, *walking*, and *talking*. The augmented dataset includes 6 action classes: *crossing*, *standing*, *queueing*, *talking*, *dancing*, and *jogging*. The *walking* action was removed from the augmented dataset because it is ill-defined [2]. We only use the bounding boxes, the associated actions, and the identities. We did not use any of the 3-D trajectory information.

Our main focus here is action recognition, and tracking is used only to improve the performance in the full model. While we show that joint optimization improves action recognition through tracking, it is intuitive that tracking performance will also improve through action recognition. However, such an evaluation is outside the scope of our work. We evaluate our results similar to [2, 3]. For each dataset, we perform a leave-one-video-out cross-validation scheme. This means that when we classify the actions in one video, we use all the other videos in the dataset for training and validation. Our action potentials are based on [12], which we also compare against to analyze the efficacy of our approach.

2.5.2 Results

Our confusion matrices for the 5-class and the 6-class datasets using the full model are shown in Figure 2.4. It is clear that removing the *walking* activity improves the classification performance, possibly due to the apparent ambiguity between *walking* and *crossing*. Our average classification accuracy is 70.9% on the

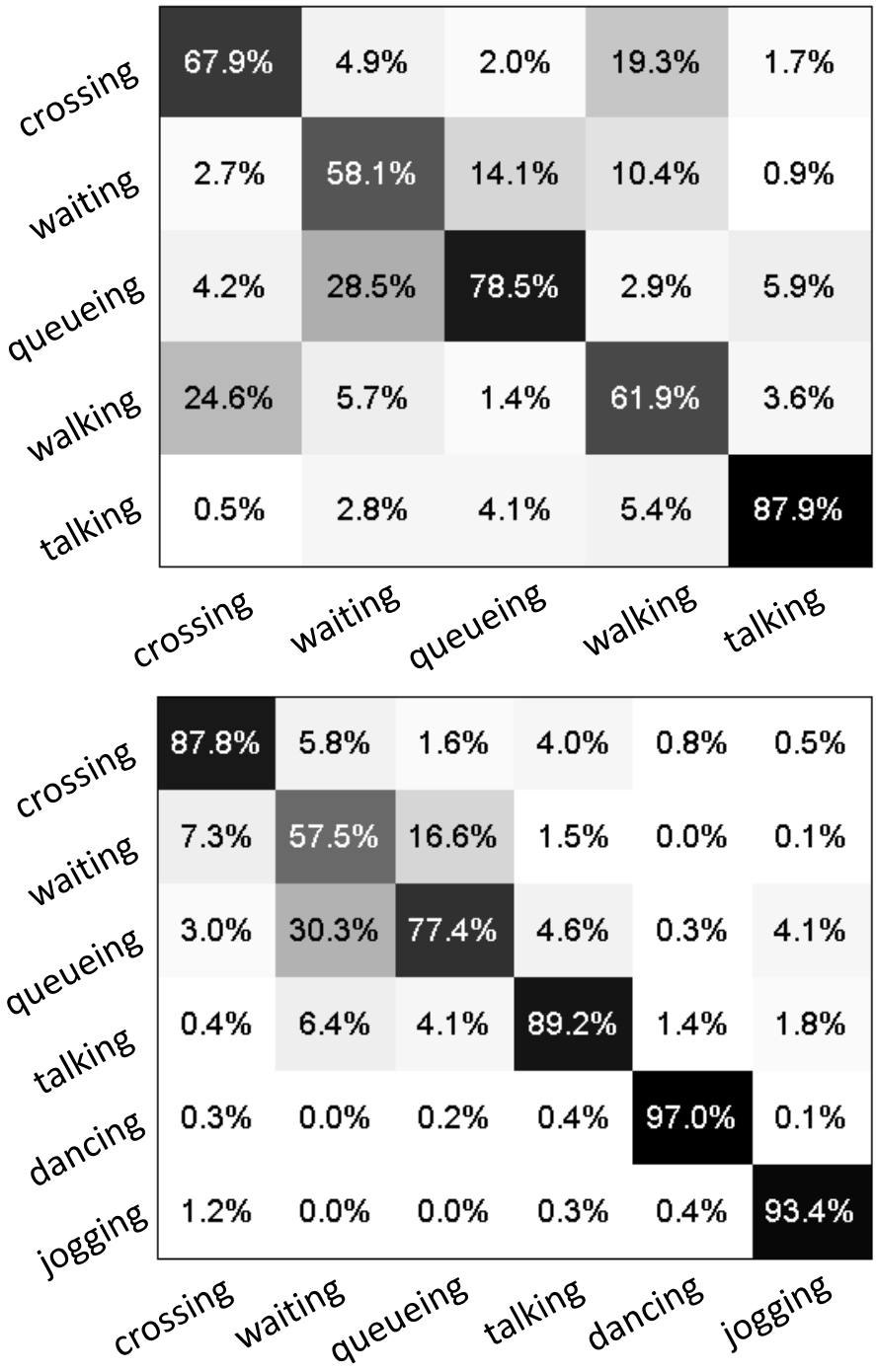

Figure 2.4: Quantitative results of our model. Our confusion matrices for the 5-class [2] and the 6-class [3] datasets.

former dataset and 83.7% on the latter.

We outperform the state-of-the-art methods on the two datasets, as shown in Table 2.1. Classification using the AC descriptor that we employ was reported in [12], which we improve upon. The model from [3] yields the same performance as our model for the first dataset. However, it employs additional trajectory information, including the 3D location and the pose of every person [3].

We also report qualitative results on the 6-activity dataset in Figure 2.5. The first two columns are the results of two consecutive frames from the same video sequence using only the action potentials, and the next two columns are the results of the same two frames, but using our full model. Each row represents a different video sequence. The first 3 sequences are successful cases where the full model improves the action classification results in an adjacent frame. The first row shows a video sequence where the misclassification of *crossing* as *queueing* is fixed with correct tracking. The second shows the same case for *talking* being misclassified as *crossing*, and the third for *jogging* being misclassified as *dancing*. The fourth row is a case where the full model actually decreases the classification accuracy due to the high confidence of the action classifier in the wrong label, causing the full model to misclassify the action in the consecutive frame.

2.6 Conclusion

We evaluated how tracking identities helps recover consistent actions across frames, and we unified action classification and identity maintenance in a single

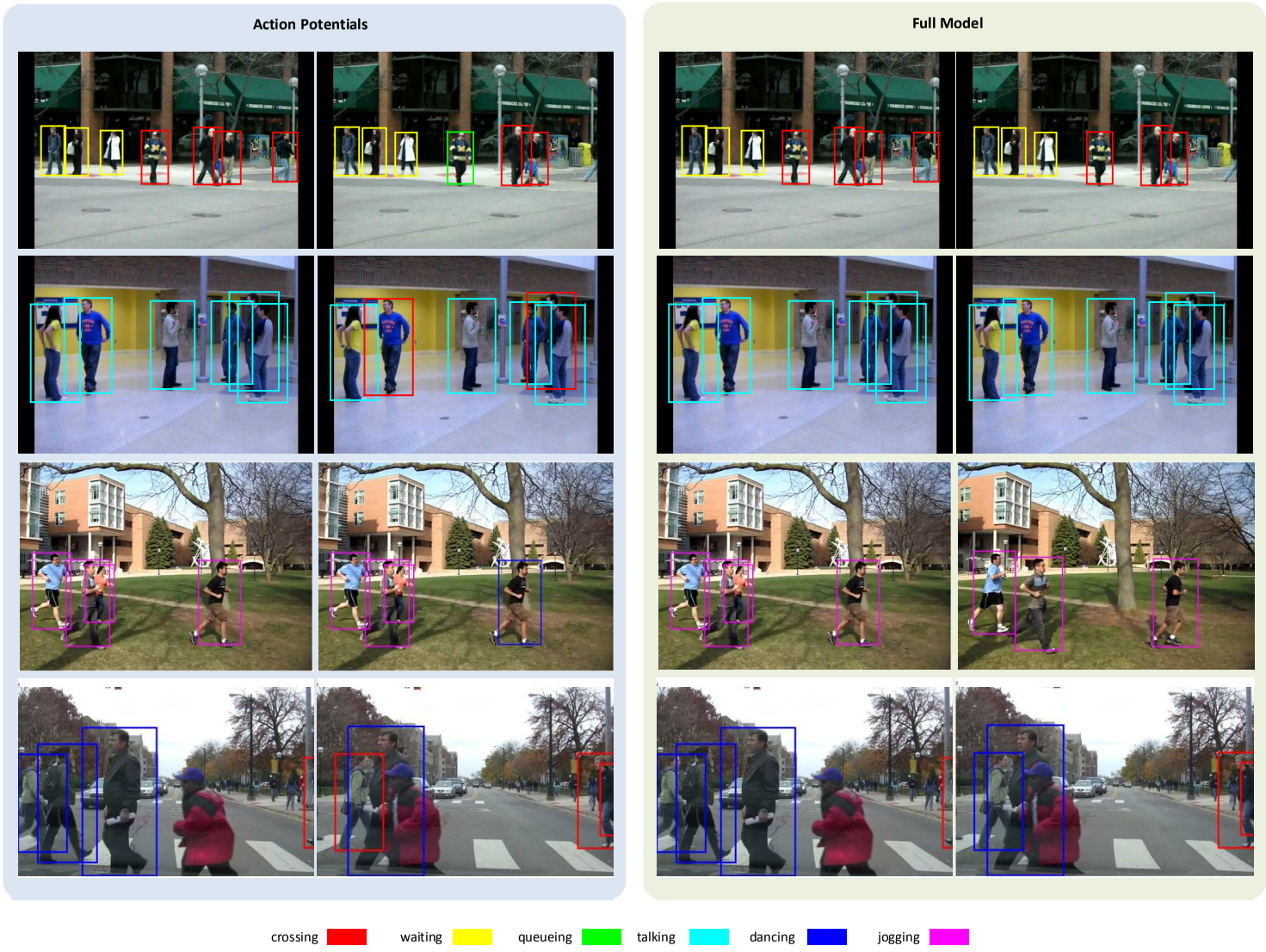

Figure 2.5: Qualitative results with and without our full model. Each row represents the result on a particular video sequence. The first 3 rows are examples where the model improves the result, while the last row is a failure case.

Approach/Dataset	5 Activities	6 Activities
AC [12]	68.2%	-
STV+MC [2]	65.9%	-
RSTV [3]	67.2%	71.7%
RSTV+MRF [3]	70.9%	82.0%
AC	68.8%	81.5%
AC+Flow	**70.9%**	**83.7%**

Table 2.1: A comparison of classification accuracies of the state-of-the-art methods on the two datasets. Our full model outperforms previous approaches and improves upon the results of the classifier output.

model. We proposed an efficient flow model to jointly solve both problems, which could be solved by a myriad of polynomial-time algorithms. In practice, we can assign actions and identities to every person in one video sequence in roughly one second. We reported our action recognition results on two datasets, and outperformed the state-of-the-art approaches using the same leave-one-out validation scheme. Our model generalizes minimum cost flow with additional constraints, and the resulting linear program is fast to optimize using off-the-shelf interior-point solvers.

Chapter 3: Combining Per-Frame and Per-Track Cues

3.1 Introduction

In this chapter we expand our temporally consistent model for human action recognition to introduce also a scene level consistency. Consider the illustration in Figure 3.1. The person outlined in the left image is queueing, while the person outlined in the right image is waiting to cross the road. Given the appearance and pose resemblance, a classifier might return similar scores for both actions for both people. However, the actions performed by the two people at a later time and the actions of people surrounding them can also provide information for the action inference task. This becomes evident when the person on the right starts crossing and nearby pedestrians start doing the same, while the person on the left stays in the queue and is surrounded by other people waiting in line; at this point, their actions become distinguishable.

Tackling this problem reveals three main challenges; action recognition, identity maintenance, and contextual harmony. We propose a representation that solves all three problems simultaneously and efficiently. A joint solution avoids the incoherences that arise from solving each problem separately. We initially train a linear SVM on the Action Context (AC) descriptor [12], which explicitly accounts for group

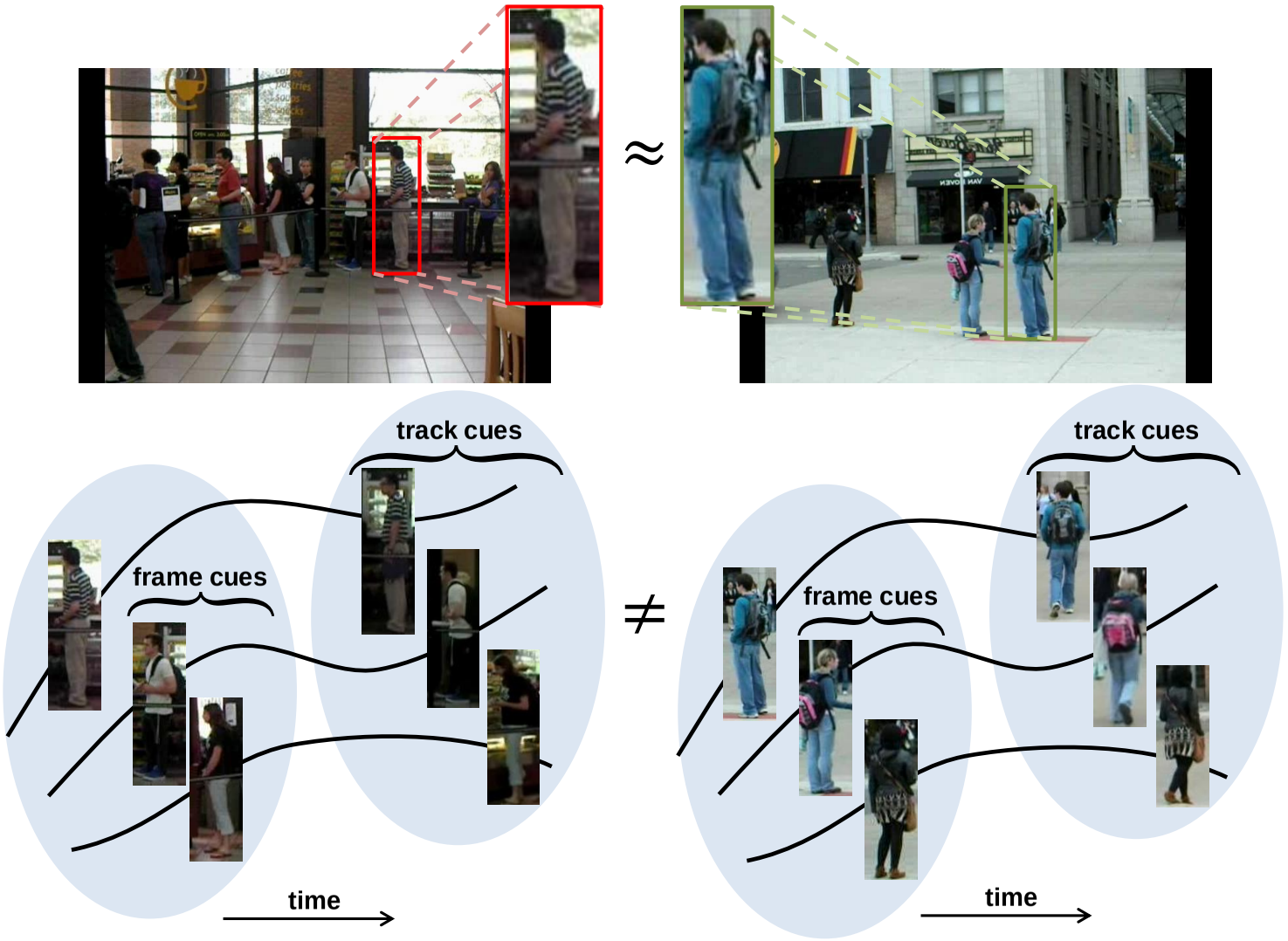

Figure 3.1: How per-frame and per-track cues can improve action recognition. By utilizing spatial and temporal cues, our joint model can overcome pitfalls an appearance-based classifier might fall into. See text for details.

actions to recognize an individual's action. We use the normalized classifier scores for the action likelihood potentials. We then train an appearance model for identity association. Our association potentials incorporate both appearance cues and action consistency cues. We also train a scene-action harmony potential, which accounts for how an action fits into the general setting of the current scene. Our problem can then be naturally represented as a constrained multi-criteria objective function. To obtain a tractable solution, we optimize this function using Dual Decomposition (or Lagrangian Relaxation) by splitting it into two subproblems, both of which are tractable and can be solved exactly and efficiently. Applied to two group action datasets, our approach outperforms state-of-the-art methods.

Our contribution in this work is three-fold:

- We propose a unified model combining per-frame and per-track cues for action recognition, solving identity maintenance in the process.
- We formulate inference as an optimization problem and solve its decompositions exactly and efficiently to recover the joint solution.
- Our action recognition performance improves upon the state-of-the-art results for two publicly available datasets.

The rest of this chapter is structured as follows. In Section 3.2 we survey the action recognition literature and discuss our contribution in its light. We introduce our approach and focus on the problem formulation in Section 3.3. We then discuss the system in details in Section 3.4. In Section 3.5, we report our quantitative and qualitative results on public datasets. And last, we conclude in Section 3.6.

3.2 Related Work

We refer the reader to the action recognition literature survey in Section 1.2, covering the recent advances in the field.

While most of these approaches require tracked to be established prior to action recognition [2, 3, 6, 7, 10, 12, 15, 16], we follow the motivation in our recent work [29], where improved results were obtained by performing identity maintenance and action recognition simultaneously and efficiently.

We also adopt the Action Context (AC) descriptor [12] and perform joint action recognition and identity maintenance, as in our prior work [29]. Additionally, we explicitly model the collective activity in a scene, its effect on individual actions, and its progression over time. Lan et al. [11] model group activities but not the temporal progression of individual actions or group activities. Also unlike [11], we do not manually specify a semantically meaningful group activity label, but instead obtain it automatically and use it only to ensure that the activities of people in the same frame are in harmony with each other. While our joint model is complex, we are still able to provide optimality and convergence guarantees without resorting to approximate inference (*e.g.*, sampling). Our approach relies on decomposing the problem into two sub-tasks, a network flow problem and a tree-structured graphical model, both of which can be solved efficiently.

3.3 Approach

3.3.1 Overview

Our focus in this work is to improve human action recognition. We assume that humans have already been localized, *e.g.*, with a state-of-the-art multi-part model [17], or with background subtraction if the camera is stationary. Our representation for a detected human figure is based on Histogram of Oriented Gradients (HOG) [22], for which we use the popular implementation from Felzenszwalb *et al.* [17]. We augment our representation with an appearance model for tracking by blurring and subsampling the three color channels of the bounding box in *Lab* color space. We use this representation to train the action and association likelihoods used in our model. We cluster the histograms of actions per-scene for our training data into a set of canonical scene types, which are then used to determine if an action is harmonious with the general setting of the current frame. We present the details of our system in the following sections.

3.3.2 Formulation

We use i, j, and k to denote the indices of human detections in a video sequence, while a, b, and c are used to denote actions. We also use f to denote frames and s to denote scenes. We define $\mathcal{P}(i)$ to be the set of candidate predecessors for human detection i from prior frames, and similarly $\mathcal{S}(i)$ to be the set of candidate successors of human detection i from subsequent frames. We also define $\mathcal{F}(i)$ to be

the frame where human detection i appears. We indicate the action and the identity of a person i by y_i and z_i, respectively, and we indicate the scene type of a frame f by q_f. We can then formulate our model as a cost function over actions, scenes, and identities represented as

$$F(\mathbf{y}, \mathbf{q}, \mathbf{z}) = \sum_f \sum_s \Bigg[g_s(f) + h_s(f) + \sum_{i \in f} \sum_a \big[u_a(i) + v'_a(i) + w_{sa}(f, i) \big] \; \mathbb{1}(y_i = a) \Bigg] \; \mathbb{1}(q_f = s), \tag{3.1}$$

where $u_a(i)$ is the classification cost associated with assigning action a to person i, $v'_a(i)$ is the associated tracking cost, and $w_{sa}(f, i)$ is the scene-action harmony cost. $g_s(f)$ denotes the scene prior cost, and $h_s(f)$ denotes the scene consistency cost. Commonly, $\mathbb{1}(.)$ is defined as the indicator function.

We define the classification cost $u_a(i)$ to be the normalized negative classification score of person i performing action a. The details of the classifier training procedure is in Section 3.4.2.

Since a detection could designate a new person entering the scene, we define our tracking cost as

$$v'_a(i) = \begin{cases} v_{ab}(i, j) & \textit{if } \exists j \in \mathcal{P}(i) \textit{ s.t. } z_i = z_j, \; y_j = b, \\ \lambda_0 & \textit{otherwise,} \end{cases} \tag{3.2}$$

where $v_{ab}(i, j)$ is the transition cost that links "person i performing action a" to a previously tracked "person j performing action b". If the newly detected person i

does not sufficiently match any of the people previously tracked, the model incurs a penalty represented by the tuning parameter λ_0, and a new track is established. We define the transition cost $v_{ab}(i,j)$ as

$$v_{ab}(i,j) = \lambda_d \; d(i,j) - \lambda_c \log(p_{ab}), \tag{3.3}$$

which is a mixture of an appearance term and an action consistency term. The appearance term measures the similarity between person i and person j with a distance metric $d(i,j)$, and the action consistency term measures the prior probability p_{ab} of a person performing action a followed by action b. The tuning parameters λ_d and λ_c weigh the importance of those two terms. The models for calculating both the appearance distance metric and the action co-occurrences are provided in Section 3.4.3.

We incorporate scene harmony by modeling a scene using the histogram of the individual actions in that scene. The scene prior cost $g_s(f)$ is calculated as the negative log prior probability p_s of the histogram of actions of scene label s. The scene consistency cost $h_s(f)$ is defined as

$$h_s(f) = \lambda_s \mathbb{1}(q_f \neq q_{f^+}), \tag{3.4}$$

where f^+ is the next frame. The scene consistency cost is in effect a smoothness prior over scenes in consecutive frames, while the scene-action harmony term $w_{sa}(f,i)$ is defined as

$$w_{sa}(f, i) = -\lambda_h \log(p_{sa}), \tag{3.5}$$

which models the likelihood p_{sa} of an individual performing action a in a scene labeled s. The tuning parameters λ_s and λ_h weigh the importance of those two terms.

We illustrate our full model in factor graph notation in Figure 3.2. The blue nodes represent human detections, the green nodes represent scenes, and the grey nodes represent the identity matching between frames. Pairwise cliques tie the scene nodes to all the detections in a specific frame, enforcing a harmonious labeling for the frame, while high-order cliques connect detections across frames to enforce both a valid identity assignment and a valid action-action transition across the tracks. Scene nodes are connected to neighboring scene nodes to discourage abrupt scene label changes.

3.3.3 Inference

Inference in our model can be formulated as a relaxed integer linear program, but it is more advantageous to leverage the underlying structure of the model. Maximum-a-posteriori (MAP) estimation in our model can be obtained using a Dual Decomposition optimization scheme [30, 31].

From Equation 3.1, our model is a function of actions, identities, and scenes. We observe that we can represent the problem via decomposition as

$$\min_{\mathbf{y},\mathbf{q},\mathbf{z}} \; F(\mathbf{y},\mathbf{q},\mathbf{z}) = \min_{\mathbf{y},\mathbf{q},\mathbf{z}} \; \big[F_1(\mathbf{y},\mathbf{q}) + F_2(\mathbf{y},\mathbf{z})\big] \tag{3.6}$$

where $F_1(.)$ is a function of the actions and scenes in each frame, while $F_2(.)$ is a function of the actions and identities across the tracks. To break the objective function into two parts, we introduce a copy of the *complicating variable* $\mathbf{y}$ for each subproblem and add a consistency (or consensus) constraint to force the two copies to match:

$$\min_{\mathbf{y}_1,\mathbf{y}_2,\mathbf{q},\mathbf{z}} \; \big[F_1(\mathbf{y}_1,\mathbf{q}) + F_2(\mathbf{y}_2,\mathbf{z})\big], \tag{3.7}$$

$$s.t. \;\; \mathbf{y}_1 = \mathbf{y}_2, \tag{3.8}$$

We now introduce the the dual variables $\boldsymbol{\nu}$ and form the Lagrangian

$$L(\mathbf{y}_1,\mathbf{y}_2,\mathbf{q},\mathbf{z},\boldsymbol{\nu}) = F_1(\mathbf{y}_1,\mathbf{q}) + F_2(\mathbf{y}_2,\mathbf{z}) + \boldsymbol{\nu}\mathbf{y}_1 - \boldsymbol{\nu}\mathbf{y}_2, \tag{3.9}$$

which can be separated into two subproblems and yields a lower bound on the optimal solution to the original problem [30]. We then form the dual problem

$$\begin{aligned} \max_{\boldsymbol{\nu}} \; & L(\mathbf{y}_1,\mathbf{y}_2,\mathbf{q},\mathbf{z},\boldsymbol{\nu}) = \\ & \max_{\boldsymbol{\nu}} \; \Bigg[\underbrace{\min_{\mathbf{y}_1,\mathbf{q}} \big[F_1(\mathbf{y}_1,\mathbf{q}) + \boldsymbol{\nu}\mathbf{y}_1\big]}_{\text{Subproblem 1}} + \underbrace{\min_{\mathbf{y}_2,\mathbf{z}} \big[F_2(\mathbf{y}_2,\mathbf{z}) - \boldsymbol{\nu}\mathbf{y}_2\big]}_{\text{Subproblem 2}} \Bigg], \end{aligned} \tag{3.10}$$

so that solving the original problem reduces to an iterative process involving the following primal-dual steps:

1. Optimize the two subproblems to obtain the primal variables $\mathbf{y}_1, \mathbf{y}_2, \mathbf{q}, \mathbf{z}$

2. Optimize the dual variables using a subgradient step $\boldsymbol{\nu} = \boldsymbol{\nu} + \eta_t \ (\mathbf{y}_1 - \mathbf{y}_2)$

where η_t is the step size for iteration t [30]. The complicating potentials in our model are the classification cost potentials $u_a(i)$ (see Figure 3.2) and therefore are distributed evenly across the two subproblems, where each subproblem is then a function of $u_a(i)/2$, for all u and i.

This approach is illustrated in Figure 3.2. This is a factor graph representation of our joint model. The blue nodes denote the human detections, the green nodes denote the scenes, and the grey nodes denote the identity matching across frames. The potentials presented in Section 3.3.2 are represented by their associated factor nodes. The decomposition described in this section is shown at the bottom, along with how the potentials (including the complicating factors) are distributed across the subproblems.

3.3.3.1 Subproblem 1.

The first subproblem is a function of the actions $\mathbf{y}$ and the scenes $\mathbf{q}$ as illustrated on the bottom left of Figure 3.2. The modified classification cost $\breve{u}_a(i)$ is defined as $u_a(i)/2+\nu_a(i)$, while the costs $g_s(f)$, $h_s(f)$, and $w_{sa}(f,i)$ are as previously defined. The problem is a tree-structured pairwise graphical model, and hence MAP inference is tractable. We optimize the subproblem exactly and efficiently by maximizing its negative objective function using Max-Product Belief Propagation [32].

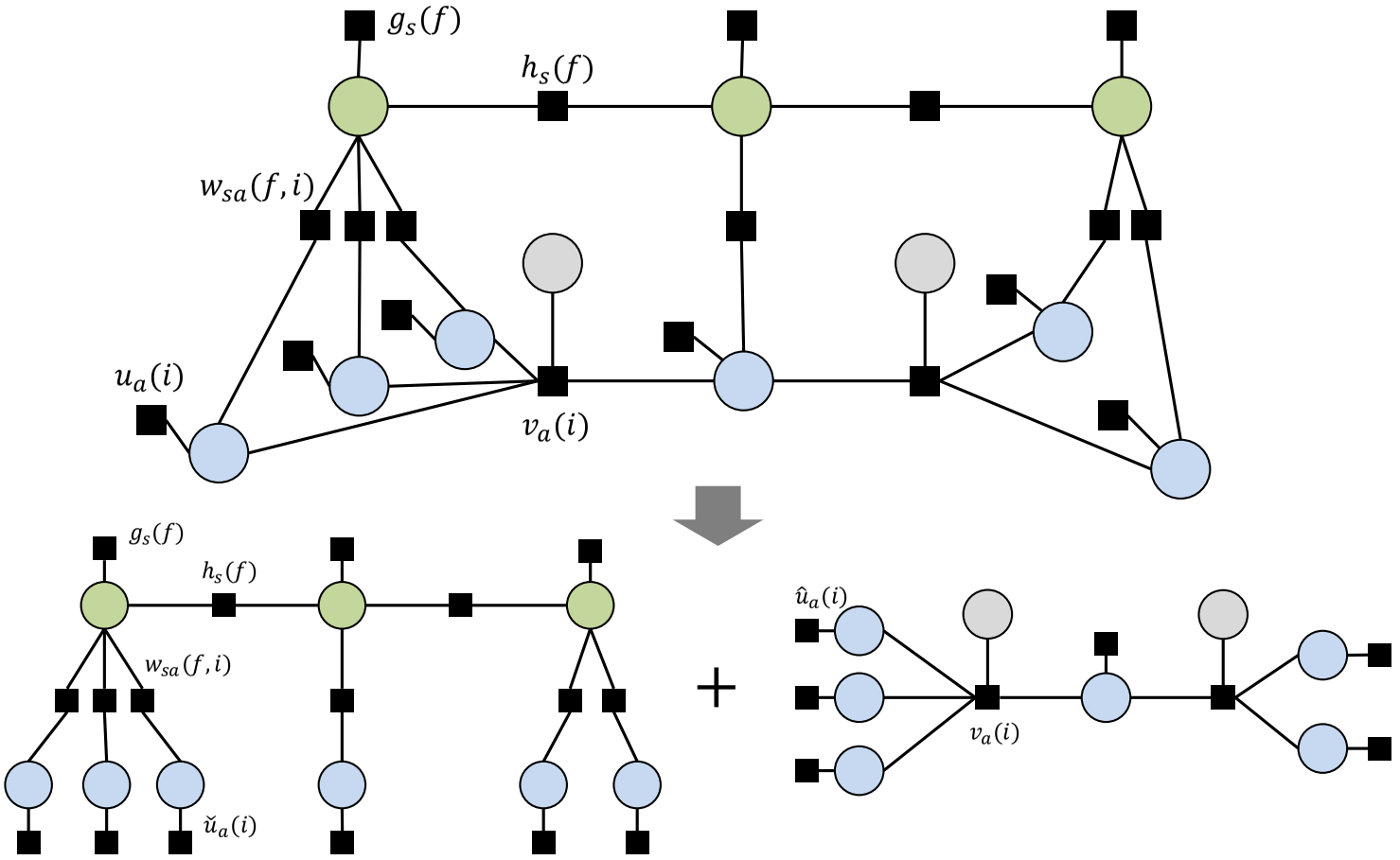

Figure 3.2: The representation of our model using factor graph notation. The figure illustrates how leveraging the underlying structure of the model can aid the inference process. Refer to the text for more details.

3.3.3.2 Subproblem 2.

The second subproblem is a function of the actions $\mathbf{y}$ and the identities $\mathbf{z}$ as illustrated on the bottom right of Figure 3.2. The high-order cliques in this problem have a special structure; they ensure the validity of the identity assignment between detections, and the consistency of actions across linked detections. While a Belief Propagation algorithm can be formulated for this problems [33], we opted to instead we use the following integer linear program (ILP) [29]

$$\begin{aligned}
\min_{\mathbf{e},\mathbf{t},\mathbf{x}} \quad & \sum_i \sum_a \Big[(\hat{u}_a(i) + \lambda_0) e_a(i) \; + \sum_{j \in \mathcal{P}(i)} \sum_b (\hat{u}_a(i) + v_{ab}(i,j)) t_{ab}(i,j) \Big], && (3.11) \\
s.t. \quad & e_a(i) \; + \sum_{j \in \mathcal{P}(i)} \sum_b t_{ab}(i,j) \; = \; x_a(i) \; + \sum_{k \in \mathcal{S}(i)} \sum_c t_{ca}(k,i) && \forall i, a \\
& \sum_a \Big[e_a(i) \; + \sum_{j \in \mathcal{P}(i)} \sum_b t_{ab}(i,j) \Big] = 1 && \forall i \\
& \{\mathbf{e}, \mathbf{t}, \mathbf{x}\} \in \mathbb{B}^n,
\end{aligned}$$

where variable $e_a(i)$ denotes the entrance of person i into the scene performing action a, while variable $t_{ab}(i,j)$ denotes the transition link of person i performing action a to person j performing action b. Finally, variable $x_a(i)$ denotes person i exiting the scene after performing action a. The entrance, transition, and exit variables are binary indicators. The cost $v_{ab}(i,j)$ is as previously defined, while the modified classification cost $\hat{u}_a(i)$ is defined as $u_a(i)/2 - \nu_a(i)$.

Minimizing the program in Equation 3.11 is equivalent to inference in the second subproblem from Equation 3.10. The form of the high-order clique potential

between detections of adjacent frames is very sparse. It does not tie the actions of everyone detected in the corresponding frame. It, however, enforces a valid match and thus a valid action transition. The variables $\mathbf{e}$, $\mathbf{t}$, and $\mathbf{x}$ always recover a unique assignment for $\mathbf{y}$ and $\mathbf{z}$. Specifically, if detection i just entered the scene, it will be assigned action $y_i = a$ for which $e_a(i) = 1$ and its identity z_i will be assigned to an unused track number. Otherwise, detection i will be instead linked to a previous detection; in that case, it will be assigned action $y_i = a$ for which $t_{ca}(k, i) = 1$ and the identity will propagate from that previous detection: $z_i = z_k$.

The ILP in Equation 3.11 represents a network flow problem [29]. In fact, the first constraint of the ILP is the "flow conservation constraint" (or *Kirchoff's Laws*). However, the second constraint, which is refered to as the "explanation constraint", is not typically encountered in the minimum cost flow problem. In this case, it enforces that an action and an identity be assigned to every person detected in the video. The flow of the minimum cost in the network uniquely assigns actions and identities to every detected person in a video sequence.

Similar to Khamis *et al.* [29], we relax the binary constraint to an interval constraint and directly solve the linear program using a fast interior-point solver. To improve the inference speed, we only connect people with overlapping bounding boxes in consecutive frames.

3.3.3.3 Solution Recovery.

On convergence, the primal variables $\mathbf{y}_1, \mathbf{y}_2, \mathbf{q}, \mathbf{z}$ and the dual variables $\boldsymbol{\nu}$ are obtained. In the case of an agreement between the two copies $\mathbf{y}_1$ and $\mathbf{y}_2$, the original complicating variable $\mathbf{y}$ is trivially recovered. Otherwise, we recover the best assignment for $\mathbf{y}$ by examining the associated dual variables $\boldsymbol{\nu}$, similar to [31]. The solution is typically attained in 3 iterations, and in several cases the global solution is attained in 6-10 iterations.

3.4 Learning

3.4.1 Piecewise Training

Since inference in our model is exact and latent variables are absent, global training approaches become not only possible, but deterministic. However, for practical reasons, we chose to use piecewise training [24]. Piecewise training involves dividing the model into several components, each of which is trained independently. We are motivated by recent theoretical and practical results. Theoretically speaking, piecewise training minimizes an upper bound on the log partition function of the model, which corresponds to maximizing a lower bound on the exact likelihood. In practice, the experiments of [24, 25] show that piecewise training sometimes outperforms global training, even when joint full inference is used. We choose to divide our model training across potentials, and train the groups of potentials independently from each other. The parameters λ_0, λ_c, λ_d, λ_s, and λ_h were manually tuned and

ultimately set to 0.25, 0.25, 0.5, 0.1, and 0.25 resepectively for all the experiments.

3.4.2 Action Potentials

We now describe how we train our action likelihood potentials. We use the AC descriptor from Lan *et al.* [12]. We employ HOG features as the underlying representation. We then train a multi-class linear SVM using *LibLinear* [26]. Next, a bag-of-words style representation for the action descriptor of each person is built. Each person is represented by the associated classifier scores, and the strongest classifier response for every action in a set of defined neighborhood regions in their context.

The descriptor of the i-th person becomes the concatenation of their action scores and context scores. The action scores for person i, given A possible actions, become $\mathbf{F}_i = [s_1(i), s_2(i), \ldots, s_A(i)]$, where $s_a(i)$ is the score of classifying person i to action a. The context score, defined over M neighborhood regions, is

$$\mathbf{C}_i = \left[\max_{j \in \mathcal{N}_1(i)} s_1(j), \ldots, \max_{j \in \mathcal{N}_1(i)} s_A(j), \ldots, \max_{j \in \mathcal{N}_M(i)} s_1(j), \ldots, \max_{j \in \mathcal{N}_M(i)} s_A(j) \right], \quad (3.12)$$

where $\mathcal{N}_m(i)$ is a list of people in the m-th region in the neighborhood of the i-th person. We use the same "sub-context regions" as [12]. We then run a second-stage classifer on the extracted AC descriptor using the same multi-class linear SVM implementation of *LibLinear* [26]. The classifier scores are negated and then normalized using a softmax function, and finally incorporated as the unary action likelihood potentials $u_a(i)$, which assign action a to person i.

3.4.3 Association Potentials

To track the identities of the targets in our video sequences, we train identity association potentials and incorporate them in our model. Our association potentials use both appearance and action consistency cues. The appearance cues are trained using the subsampled color channels as features. We train for a Mahalanobis distance matrix M to estimate the similarity between detections across frames. The distance matrix is learned so as to bring detections from the same track closer, and those from different tracks apart [27]. This is formulated as

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \sum_{\mathcal{T}_k} \Bigg[\sum_{i,j \in \mathcal{T}_k} (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j) \; - \sum_{\substack{i' \in \mathcal{T}_k \\ j' \notin \mathcal{T}_k}} (\mathbf{f}_{i'} - \mathbf{f}_{j'})^T \mathbf{M} (\mathbf{f}_{i'} - \mathbf{f}_{j'}) \Bigg], \quad (3.13)$$

where $\mathcal{T}_k$ is the k-th track and $\mathbf{f}_i$ is the feature vector of the i-th person. We solve for M using the fast Large Margin Nearest Neighbor (LMNN) implementation of [28]. The distance between the features of two detected people i and j can then be defined as

$$d(i,j) = (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j). \quad (3.14)$$

The action consistency cues are estimated using the groundtruth action labels from the training set. We count pairwise co-occurrences of actions on the same track plus a small additive smoothing parameter α. The counts are normalized into the pairwise co-occurrence probabilities p_{ab} of action pairs a and b.

3.4.4 Scene Potentials

We cluster the histograms of actions in all the frames of our training set using k-means, where we set $k = 8$ in all of our experiments. The k-means cluster centroids are good representatives of the most likely scenes, and so the centroid histograms are an appropriate approximation for the likelihood of an action given a scene canonical scene types, while the number of points in each cluster is used to approximate the scene prior probability. The form of our scene potentials is similar to the harmony potentials introduced in [34], but our training approach is different.

3.5 Experiments

3.5.1 Datasets

We use the group actions dataset from [2] and its augmentation from [3] to evaluate our model. The datasets are appropriate since they have multiple targets in a natural setting, while most action datasets, like KTH [5] or Weizmann [4], have a single person performing a specific action. The original dataset includes 5 action classes: *crossing*, *standing*, *queueing*, *walking*, and *talking*. The augmented dataset includes 6 action classes: *crossing*, *standing*, *queueing*, *talking*, *dancing*, and *jogging*. The *walking* action was removed from the augmented dataset because it is ill-defined [2]. We only use the bounding boxes, the associated actions, and the identities. We did not use any of the 3-D trajectory information.

Our main focus here is action recognition, and tracking is used only to improve

the performance in the full model. We evaluate our results similar to [2, 3]. For each dataset, we perform a leave-one-video-out cross-validation scheme. This means that when we classify the actions in one video, we use all the other videos in the dataset for training and validation. Our action potentials are based on [12], which we also compare against to analyze the efficacy of our approach.

3.5.2 Results

Our confusion matrices for the 5-class and the 6-class datasets using the full model are shown in Figure 3.3. It is clear that removing the *walking* activity improves the classification performance, possibly due to the apparent ambiguity between *walking* and *crossing*. Our average classification accuracy is 72.0% on the former dataset and 85.8% on the latter.

We outperform the state-of-the-art methods on the two datasets, as shown in Table 3.1. Classification using the AC descriptor that we employ was reported in [12], which we improve upon. Our full model outperforms previous approaches and can be solved deterministically with some global optimality guarantees. It is worth noting that the model from [3] employs additional trajectory information, including the 3D location and the pose of every person [3].

We also report qualitative results on the 6-activity dataset in Figure 3.4. The four columns represent the results using our unary potentials only, the track cues, the frame cues, and the full model respectively. The first row is a case where the full model, combining both cues, outperforms using either the frame cues or the

Approach/Dataset	5 Activities	6 Activities
AC [12]	68.2%	-
STV+MC [2]	65.9%	-
RSTV [3]	67.2%	71.7%
RSTV+MRF [3]	70.9%	82.0%
Unary (AC) [29]	68.8%	81.5%
AC+Track Cues [29]	70.9%	83.7%
AC+Frame Cues	70.7%	84.8%
AC+Full Model	**72.0%**	**85.8%**

Table 3.1: A comparison of classification accuracies of the state-of-the-art methods on the two datasets. Our full model outperforms previous approaches and can be solved deterministically with some global optimality guarantees.

track cues in isolation. In the second row the track cues degraded the results of the unary potentials due to identity matching inaccuracies in the busy scene, but the full model still yielded a perfect classification result. The frame cues were not able to fix classifier errors in the third row, but the full model leveraged tracking and reported accurate results. Finally, the final row is a failure case where, through the high classifier confidence in the wrong label, the full model reinforced the wrong result, classifying everyone incorrectly, even though the frame cues were successful.

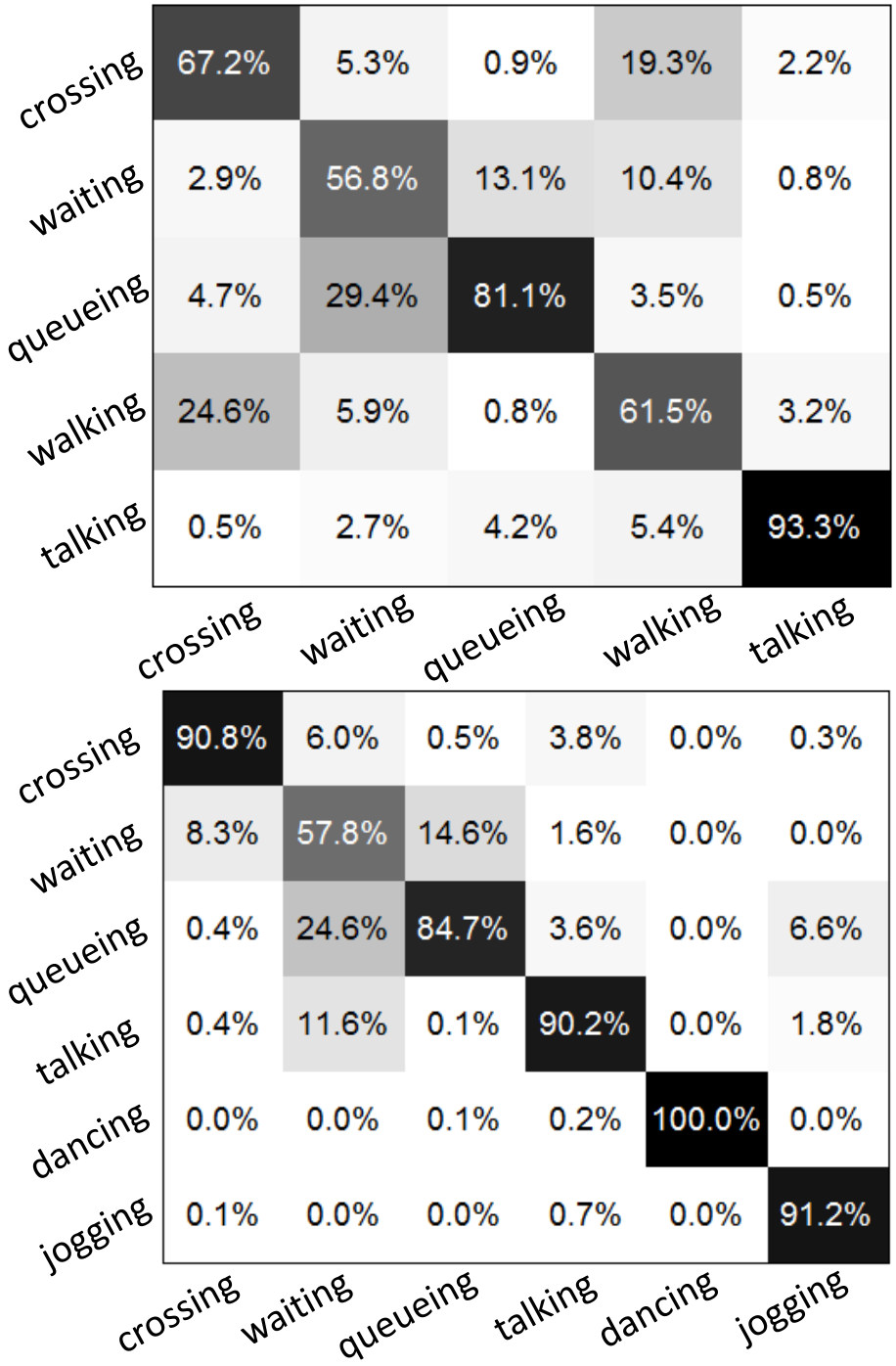

Figure 3.3: Quantitative results of our model. Our confusion matrices for the 5-class [2] and the 6-class [3] datasets.

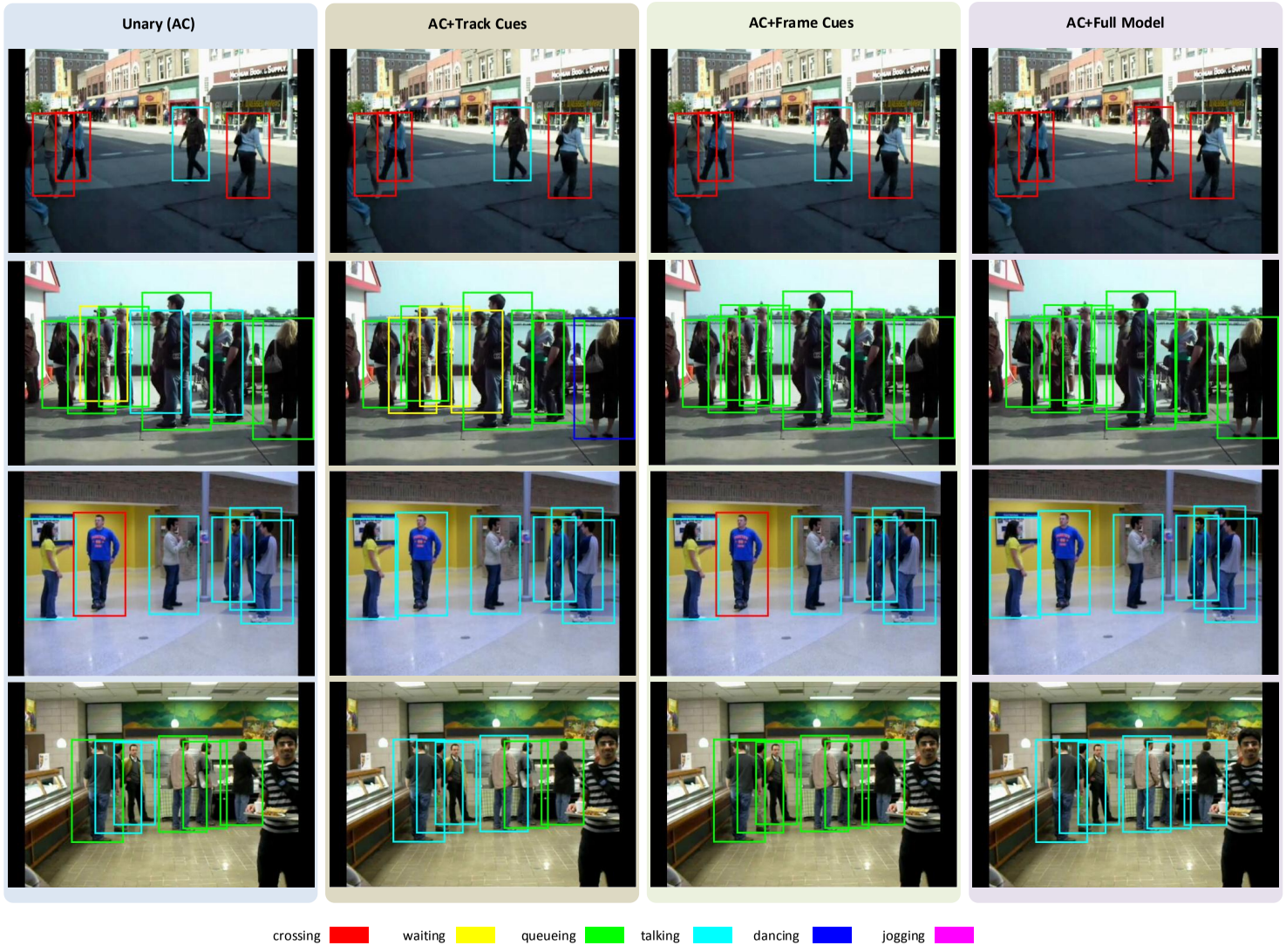

Figure 3.4: Qualitative results of our model. Each row in the figure represents a different video sequence. Each row represents the result on a particular video sequence. The first 3 rows are examples where the model improves the result, while the last row is a failure case.

3.6 Conclusion

We introduced a model that combines tracking cues and scene cues to improve action classification results. The intractability of our model is overcome by a decomposition that leverages its underlying structure. The decomposition yields two subproblems, which we solve exactly and efficiently. We recover the solution to the original problem, which is optimal in several cases. Finally, by combining both cues, we reported action recognition results that outperform the state-of-the-art on two publicly available datasets using the same validation scheme.

Chapter 4: Probabilistic Logic for Collective Activity Recognition

4.1 Introduction

In many computer vision tasks, it is useful to combine structured, high-level reasoning with low-level predictions. Collective reasoning at a high-level can take advantage of accurate low-level detectors, while improving the accuracy of predictions based on less accurate detectors. To fully leverage the power of high-level reasoning, we require a tool that is both powerful enough to model complex, structured problems and expressive enough to easily encode high-level ideas. In this chapter we apply *hinge-loss Markov random fields* (HL-MRFs) [35, 36] to our task of interest, human activity recognition from videos. HL-MRFs are powerful, templated graphical models that admit efficient, exact inference over continuous variables. We demonstrate that, when combined with the modeling language *probabilistic soft logic* (PSL) [37,38], HL-MRFs allow us to design high-level, structured models that improve the performance of low-level detectors.

We focus on the task of collective *activity recognition* of humans in video scenes. Since human activities are often interactive or social in nature, collective reasoning over activities can provide more accurate detections than independent, local predictions. For instance, one can use aggregate predictions within the scene

or frame to reason about the local actions of each actor. Further, collective models let us reason across video frames, to allow predictions in adjacent frames to inform each other, and thus implement the intuition that actions are temporally continuous.

We demonstrate the effectiveness of HL-MRFs and PSL on two group activity datasets. Using a simple, interpretable model, we are able to achieve significant lift in accuracy from low-level predictors. We thus show HL-MRFs to be a powerful, expressive tool for high-level computer vision.

4.1.1 Related Work

Section 1.2 provides a survey of the recent developments of action recognition research in recent years.

We formulate a powerful approach to model the complex and rich structure in action recognition, going beyond the independent classifications resulting from low level detectors. Recent work in multi-person action recognition carried a similar motivation. We presented a network flow model to perform simultaneous action recognition and identity maintenance [29]. We then augmented that model to jointly reason about scene types [39]. Similarly, Choi *et al*. proposed a unified model to perform action recognition at the individual and group levels simultaneously with tracking [40]. We build upon this work using a probabilistic relational approach.

PSL is one of many existing systems for probabilistic relational modeling, including *Markov logic networks* [41], *relational dependency networks* [42], and *relational Markov networks* [43], among others. One distinguishing feature of PSL is

that its continuous representation of logical truth makes its underlying probabilistic model an HL-MRF [36], which allows inference of the *most-probable explanation* (MPE) to be solved as a convex optimization. Our work benefits from recent advances on fast HL-MRF inference based on the *alternating direction method of multipliers* [35,44], which significantly increases the scalability of HL-MRF inference over off-the-shelf convex optimization tools.

4.2 Hinge-loss Markov Random Fields

In this section we formally introduce *hinge-loss Markov random fields* (HL-MRFs), a general class of conditional, continuous-valued probabilistic models. HL-MRFs are log-linear probabilistic models whose features are hinge-loss functions of the variable states. Through constructions based on *soft logic* (explained in Section 4.3), hinge-loss potentials can be used to model generalizations of logical conjunction and implication, making these powerful models interpretable, flexible, and expressive.

HL-MRFs are parameterized by constrained hinge-loss energy functions.

Definition 1. *Let* $\mathbf{Y} = (Y_1, \ldots, Y_n)$ *be a vector of* n *variables and* $\mathbf{X} = (X_1, \ldots, X_{n'})$ *a vector of* n' *variables with joint domain* $\mathbf{D} = [0,1]^{n+n'}$*. Let* $\phi = (\phi_1, \ldots, \phi_m)$ *be* m *continuous potentials of the form*

$$\phi_j(\mathbf{Y}, \mathbf{X}) = [\max\{\ell_j(\mathbf{Y}, \mathbf{X}), 0\}]^{p_j}$$

where ℓ_j *is a linear function of* $\mathbf{Y}$ *and* $\mathbf{X}$ *and* $p_j \in \{1, 2\}$*. Let* $C = (C_1, \ldots, C_r)$ *be linear constraint functions associated with index sets denoting equality constraints* $\mathcal{E}$

and inequality constraints $\mathcal{I}$, which define the feasible set

$$\tilde{\mathbf{D}} = \left\{ \mathbf{Y}, \mathbf{X} \in \mathbf{D} \middle| \begin{array}{l} C_k(\mathbf{Y}, \mathbf{X}) = 0, \forall k \in \mathcal{E} \\ \\ C_k(\mathbf{Y}, \mathbf{X}) \geq 0, \forall k \in \mathcal{I} \end{array} \right\}.$$

For $\mathbf{Y}, \mathbf{X} \in \tilde{\mathbf{D}}$, given a vector of nonnegative free parameters, i.e., weights, $\lambda = (\lambda_1, \ldots, \lambda_m)$, a constrained hinge-loss energy function *f_λ is defined as*

$$f_\lambda(\mathbf{Y}, \mathbf{X}) = \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{Y}, \mathbf{X}) \ .$$

Definition 2. *A* hinge-loss Markov random field *P over random variables $\mathbf{Y}$ and conditioned on random variables $\mathbf{X}$ is a probability density defined as follows: if $\mathbf{Y}, \mathbf{X} \notin \tilde{\mathbf{D}}$, then $P(\mathbf{Y}|\mathbf{X}) = 0$; if $\mathbf{Y}, \mathbf{X} \in \tilde{\mathbf{D}}$, then*

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\lambda)} \exp\left[-f_\lambda(\mathbf{Y}, \mathbf{X})\right] \ , \tag{4.1}$$

where $Z(\lambda) = \int_{\mathbf{Y}} \exp\left[-f_\lambda(\mathbf{Y}, \mathbf{X})\right]$.

The potential functions and weights can be grouped together into *templates*, which are used to define general classes of HL-MRFs that are parameterized by the structure of input data. Let $\mathcal{T} = (t_1, \ldots, t_s)$ denote a vector of templates with associated weights $\Lambda = (\Lambda_1, \ldots, \Lambda_s)$. We partition the potentials by their associated templates and let

$$\Phi_q(\mathbf{Y}, \mathbf{X}) = \sum_{j \in t_q} \phi_j(\mathbf{Y}, \mathbf{X})$$

for all $t_q \in \mathcal{T}$. In the *ground* HL-MRF, the weight of the j'th hinge-loss potential is set to the weight of the template from which it was derived, i.e., $\lambda_j = \Lambda_q$, for each $j \in t_q$.

MPE inference in HL-MRFs is equivalent to finding the feasible minimizer of the convex energy f_λ. Here, HL-MRFs have a distinct advantage over general discrete models, since minimizing f_λ is a convex optimization, rather than a combinatorial one.

Bach *et al.* showed how to minimize f_λ using a consensus-optimization algorithm [35], based on the *alternating direction method of multipliers* (ADMM) [44]. Consensus-optimization works by creating local copies of the variables in each potential and constraint, constraining them to be equal to the original variables, and relaxing those equality constraints to make independent subproblems. By iteratively solving the subproblems and averaging the results, the algorithm reaches a consensus on the best values of the original variables, also called the *complicating variables* or the *consensus variables*. This procedure is guaranteed to converge to the global minimizer of f_λ [44].

The inference algorithm has since been generalized and improved [36]. Experimental results suggest that the running time of this algorithm scales linearly with the size of the problem. On modern hardware, the algorithm can perform exact MPE inference with hundreds of thousands of variables in just a few seconds.

4.2.1 Weight Learning

To learn the parameters Λ of an HL-MRF given a set of training examples, we perform *maximum-likelihood estimation* (MLE), using the *voted perceptron* algorithm [45]. The partial derivative of the log of Equation 4.1 with respect to a

parameter Λ_q is

$$\frac{\partial \log P(\mathbf{Y}|\mathbf{X})}{\partial \Lambda_q} = \mathbb{E}_\Lambda \left[\Phi_q(\mathbf{Y}, \mathbf{X})\right] - \Phi_q(\mathbf{Y}, \mathbf{X}) \,, \tag{4.2}$$

where $\mathbb{E}_\Lambda$ is the expectation under the distribution defined by Λ. Note that the expectation in Equation 4.2 is intractable to compute. To circumvent this, we use a common approximation: the values of the potential functions at the most probable setting of $\mathbf{Y}$ with the current parameters [37]. The MPE approximation of the expectation is fast, due to the speed of the inference algorithm; however, there are no guarantees about its quality.

The voted perceptron algorithm optimizes Λ by taking steps of fixed length in the direction of the negative gradient, then averaging the points after all steps. To preserve the non-negativity of the weights, any step that is outside the feasible region is projected back before continuing. For a smoother ascent, it is often helpful to divide the q-th component of the gradient by the number of groundings $|t_q|$ of the q'th template [46], which we do in our experiments.

4.3 Probabilistic Soft Logic

In this section, we review *probabilistic soft logic* (PSL) [37, 38], a declarative language for probabilistic reasoning. While PSL borrows the syntax of first-order logic, semantically, all variables take *soft truth values* in the interval $[0, 1]$, instead of only the extremes, 0 (FALSE) and 1 (TRUE). Continuous variables are useful both for modeling continuous domains as well as for expressing confidences in discrete predictions, which are desirable for the same reason that practitioners often prefer

marginal probabilities to discrete MPE predictions. PSL provides a natural interface to design hinge-loss potential templates using familiar concepts from first-order logic.

A PSL program consists of a set of first-order logic rules with conjunctive bodies and disjunctive heads. Rules are constructed using the logical operators for conjunction ($\wedge$), negation ($\neg$) and implication ($\Rightarrow$).

Rules are assigned weights, which can be learned from observed data. Consider the following rule for collective image segmentation.

$$0.8 : \textsc{Close}(P_1, P_2) \wedge \textsc{Label}(P_1, C) \Rightarrow \textsc{Label}(P_2, C)$$

In this example, P_1, P_2 and C are variables representing two pixels and a category; the predicate $\textsc{Close}(P_1, P_2)$ measures the degree to which P_1, P_2 are "close" in the image; $\textsc{Label}(P_1, C)$ indicates the degree to which P_1 belongs to class C, and similarly for $\textsc{Label}(P_2, C)$. This rule has weight 0.8.

PSL uses the *Lukasiewicz t-norm*, and its corresponding *co-norm*, to relax the logical operators for continuous variables. These relaxations are exact at the extremes, but provide a consistent mapping for values in between. For example, given variables X and Y, the relaxation of the conjunction $X \wedge Y$ would be $\max\{0, X + Y - 1\}$.

We say that a rule r is *satisfied* when the truth value of the head r_{HEAD} is at least as great as that of the body r_{BODY}.

The rule's *distance from satisfaction* d_r measures the degree to which this condition is violated:

$$d_r = \max\{0, r_{\text{BODY}} - r_{\text{HEAD}}\}. \tag{4.3}$$

This corresponds to one minus the truth value of $r_{\text{BODY}} \Rightarrow r_{\text{HEAD}}$ when the variables are $\{0, 1\}$-valued. In the process known as *grounding*, each rule is instantiated for all possible substitutions of the variables as given by the data. For example, the above rule would be grounded for all pairs of pixels and categories.[1]

Notice that Equation 4.3 corresponds to a convex hinge function. In fact, each rule corresponds to a particular template $t \in \mathcal{T}$, and each grounded rule corresponds to a potential in the ground HL-MRF. If we let $\mathbf{X}_{i,j}$ denote the closeness of pixels p_i, p_j, and $Y_{i,c}$ denote the degree to which p_i has label c (likewise for p_j), then the example rule above would correspond to the potential function

$$\phi(\mathbf{Y}, \mathbf{X}) = [\max\{0, X_{i,j} + Y_{i,c} - Y_{j,c} - 1\}]^p,$$

where $p \in \{1, 2\}$ is the exponent parameter (see Definition 1). Thus, PSL, via HL-MRFs, defines a log-linear distribution over possible interpretations of the first-order rules.

Because it is backed by HL-MRFs, PSL has some additional features that are useful for modeling. The constraints in Definition 1 allow the encoding of functional modeling requirements, which can be used to enforce mutually exclusion constraints (i.e., that the soft-truth values should sum to one). Further, the exponent parameter p allows flexibility in the shape of the hinge, affecting the sharpness of the penalty for violating the logical implication. Setting p to 1 penalizes violation linearly with the amount the implication is unsatisfied, while setting p to 2 penalizes small violations

[1]Though this could possibly lead to an explosion of groundings, PSL uses lazy activation to only create groundings for substitutions when the truth value of the body exceeds a certain margin.

much less. In effect, some linear potentials overrule others, while the influences of squared potentials are averaged together.

4.4 Collective Activity Recognition

In this section, we apply HL-MRFs to the task of collective activity recognition. We treat this as a high-level vision task, using the output of primitive, local models as input to a collective model for joint reasoning. We begin by describing the datasets and objective. We then describe our model. We conclude with a discussion of our experimental results.

4.4.1 Datasets

We use the collective activity dataset from [2] and its augmentation from [3] to evaluate our model. The first dataset contains 44 video sequences, each containing multiple actors performing activities in the set: *crossing*, *standing*, *queueing*, *walking*, and *talking*. The second dataset contains 63 sequences, with actions in: *crossing*, *standing*, *queueing*, *talking*, *dancing*, and *jogging*.[2] From each dataset, we use the bounding boxes (with position, width and height), pixel data, actions and identity annotations; we do not use the 3-D trajectories. Activity recognition in these datasets is challenging, since the scenes involve multiple actors in a natural setting; other action datasets, like KTH [5] or Weizmann [4], have a single person performing a specific action. In addition, there is considerable ambiguity in the ac-

[2]The *walking* action was removed from the augmented dataset by [3] because it was deemed ill-defined.

tions being considered; for example, the actions *standing* and *queueing* are difficult to distinguish, even for a human. Figure 4.1 illustrates some sample frames from the two datasets. The original dataset and its augmentation include multiple actors in a natural setting performing specific actions. The colors of the bounding boxes in the figure specify the groundtruth action of the corresponding person.

Similar to our prior work [29, 39], we represent the detected human figures using *histogram of oriented gradients* (HOG) [22] features and *action context* (AC) descriptors [12]. The AC descriptor is a feature representation that combines the local beliefs about an actor's activities with those of actors in surrounding spatiotemporal neighborhoods. To create the AC descriptors, we use HOG features as the underlying feature representation; we then train a first-level SVM classifier on these features and combine the outputs per [12]. Finally, we train a second-stage SVM classifier on the AC descriptors to obtain the activity beliefs used in our high-level model. All classifiers are trained using a leave-one-out methodology, such that the predictions for the i'th sequence are obtained by training on all other sequences.

4.4.2 Model

Our primary objective is to enhance the low-level activity detectors with high-level, global reasoning. To do so, we augment the local features (described below) using relational information within and across adjacent frames.

By modeling the relationships of bounding boxes, we can leverage certain intuitions about human activity. For instance, it is natural to assume that one's

Figure 4.1: A few sample frames from the collective activity datasets. The original dataset and its augmentation contain multiple actors in various scenes. The bounding box colors specify the groundtruth label for the action of each person.

activity is temporally continuous; that is, it is not likely to change between points close in time. Further, there are certain activities that involve interaction with others, such as *talking* or *queueing*. Therefore, if we believe that one or more actors are talking, then actors nearby are also likely to be talking. Using PSL, modeling these intuitions is a simple matter of expressing them in first-order logic. We can then use HL-MRFs to reason *jointly* over these rules.

Our PSL model is given below.

$$\textsc{Local}(B, a) \Rightarrow \textsc{Doing}(B, a) \tag{R1}$$

$$\textsc{Frame}(B, F) \wedge \textsc{FrameLabel}(F, a) \Rightarrow \textsc{Doing}(B, a) \tag{R2}$$

$$\textsc{Close}(B_1, B_2) \wedge \textsc{Doing}(B_1, a) \Rightarrow \textsc{Doing}(B_2, a) \tag{R3}$$

$$\textsc{Seq}(B_1, B_2) \wedge \textsc{Close}(B_1, B_2) \Rightarrow \textsc{Same}(B_1, B_2) \tag{R4}$$

$$\textsc{Same}(B_1, B_2) \wedge \textsc{Doing}(B_1, a) \Rightarrow \textsc{Doing}(B_2, a) \tag{R5}$$

Rule R1 corresponds to beliefs about local predictions (on either the HOG features or AC descriptors). R2 expresses the belief that if many actors in the current frame are doing a particular action, then perhaps everyone is doing that action. To implement this, we derive a FrameLabel predicate for each frame; this is computed by accumulating and normalizing the Local activity beliefs for all actors in the frame. Similarly, R3 enforces our intuition about the effect of proximity on activity, where actors that are close[3] in the same frame are likely to perform the same action. This can be considered a fine-grained version of the second rule. R4 is used for identity maintenance and tracking. It essentially says that if two bounding

[3]To measure closeness, we use an RBF kernel.

boxes occur in adjacent frames and their positions have not changed significantly, then they are likely the same actor. We then reason, in R5, that if two bounding boxes (in adjacent frames) refer to the same actor, then they are likely to be doing the same activity. Note that rules involving lowercase a are defined for each action a, such that we can learn different weights for different actions. We define priors over the predicates SAME and DOING, which we omit for space. We also define (partial) functional constraints (not shown), such that the truth-values over all actions (respectively, over all adjacent bounding boxes), sum to (at most) one. We train the weights for these rules using 50 iterations of voted perceptron, with a step size of 0.1.

Note that we perform identity maintenance only to improve our activity predictions. During prediction, we do not observe the SAME predicate, so we have to predict it. We then use these predictions to inform the rules pertaining to activities.

4.4.3 Experiments

To illustrate the lift one can achieve on low-level predictors, we evaluate two versions of our model: the first uses activity beliefs from predictions on the HOG features; the second uses activity beliefs predicted on the AC descriptors. Essentially, this determines which low-level predictions are used in the predicates LOCAL and FRAMELABEL. We denote these models by HL-MRF + HOG and HL-MRF + AC respectively. We compare these to the predictions made by the first-stage predictor (HOG) and the second-stage predictor (AC).

Method	5 Activities		6 Activities	
	Acc.	F1	Acc.	F1
HOG	.474	.481	.596	.582
HL-MRF + HOG	**.598**	**.603**	**.793**	**.789**
AC	.675	.678	.835	.835
HL-MRF + AC	**.692**	**.693**	**.860**	**.860**

Table 4.1: Results of experiments with the 5- and 6-activity datasets, using leave-one-out cross-validation. Scores are reported as the cumulative accuracy/F1, to account for size and label skew across folds.

The results of these experiments are listed in Table 4.1. We also provide recall matrices (row-normalized confusion matrices) for HL-MRF + AC in Figure 4.2. For each dataset, we use leave-one-out cross-validation, where we train our model on all except one sequence, then evaluate our predictions on the hold-out sequence. We report cumulative accuracy and F1 to compensate for skew in the size and label distribution across sequences; this involves accumulating the confusion matrices across folds.

Our results illustrate that our models are able to achieve significant lift in accuracy and F1 over the low-level detectors. Specifically, we see that HL-MRF + HOG achieves a 12 to 20 point lift over the baseline HOG model, and HL-MRF + AC obtains a 1.5 to 2.5 point lift over the AC descriptor.

4.5 Conclusion

We have shown that HL-MRFs are a powerful class of models for high-level computer vision tasks. When combined with PSL, designing probabilistic models is easy and intuitive. We applied these models to the task of collective activity recognition, building on local, low-level detectors to create a global, relational model. Using simple, interpretable first-order logic rules, we were able to improve the accuracy of low-level detectors.

Figure 4.2: Quantitative results on the two datasets. We show the recall matrices (i.e., row-normalized confusion matrices) for the 5- and 6-activity datasets, using the HL-MRF + AC model.

Chapter 5: Multi-Label Action Recognition

5.1 Introduction

As previously mentioned, recent work in action recognition research has gone beyond the classic isolated action short video [4, 5] to incorporate hierarchical, contextual, interactional, and spatio-temporal cues [6–8, 39, 40]. However, what all these approaches have in common is that they assume that action recognition is a multi-class problem, where only the most probable label for each actor is predicted.

Multi-class classification is a fundamental problem in machine learning. For many approaches, training is performed in a one-vs-all fashion, where instances from one class are set as positive and the rest negative. Test instances are evaluated and assigned to the class with the highest score. This is appropriate for many problems where labels are mutually exclusive. In semantic segmentation, for instance, each pixel is assigned the name of the class it belongs to. Given that each pixel maps to a single object, and assuming the list of classes do not overlap, multi-class classifications is a natural formulation for the problem [25]. However, if the question we are interested in is "What are they doing?" [2], assigning each actor a single label seems unnecessarily limiting.

Consider the sample frame from the Collective Activity dataset [2] in Fig-

ure 5.1. The two actors in the frame are talking while standing in line, two naturally co-occurring actions. The groundtruth labels for both, however, is the single label *queueing*. In the multi-class setting where a classifier is accordingly only allowed a single label to choose, assigning the label *talking* or *waiting* to either actor is an error and a False Positive for the *talking* or the *waiting* classifier. On the other hand, knowing that the labels *talking*, *queueing*, and *waiting* strongly correlate, a multi-label approach would likely assign the three correct labels to both actors. On the other hand, inversely correlated actions like *queueing* and *crossing* are unlikely to be assigned at the same time to an actor. While the dataset strongly motivated our work, it was not a suitable candidate for our experiments because the actors in most videos were performing the same action.

We propose to treat action recognition as a multi-label classification problem. Each actor can be assigned a subset of the power set of action labels. One can pose multi-label classification as multi-class classification with an exponential number of classes, where each subset of the power set is a separate class. This formulation, however, is computationally infeasible. Equally difficult to solve is formulating multi-label classification as structured prediction for a densely connected Markov Random Field (MRF) of labels, where inference is generally intractable, and typically approaches resort to restricting the structure of the MRF to a tree or at least to small tree width. Instead, we extend recent work on multi-label classification with densely correlated labels [1]. However, instead of assuming an apriori known correlation matrix, we formulate both problems - multi-label training and label correlation estimation - as a joint max-margin bilinear optimization problem. This has

Figure 5.1: The case for multi-label action recognition. People in natural settings perform more than one action at the same time. Our approach takes into account pairwise correlations to ensure assigned action combinations are meaningful.

the advantage that both problems are optimized to jointly minimize an appropriate loss on the training set. Additionally, discriminatively learning both the classifiers and the label correlations is empirically shown to yield classifiers with better performance accuracy. Finally, given the lack of datasets for our task, we relabeled the UCLA Courtyard dataset [47] using the same set of labels, but instead each actor is assigned a subset of labels instead of a single label.

Our main contribution here is three-fold:

- We recast action recognition to the multi-label setting. While attributes, inherently multi-label, have been leveraged before in action recognition to describe the action, the human body configuration, or the manipulated objects, the action recognition problem in itself has always been treated as a multi-class problem.

- We introduce a bilinear classification approach where we jointly and discriminatively learn both the classifiers and the label correlations, generalizing previous work where the label correlations were considered prior knowledge or estimated offline.

- Finally, we relabeled the UCLA Courtyard dataset to be the first multi-label action recognition benchmark.

The rest of this chapter is organized as follows. The action and activity recognition literature is surveyed in Section 5.2. We introduce our joint formulation for multi-label training and correlation estimation in Section 5.3, and we propose an algorithm to efficiently optimize it. We then present the relabeled UCLA Courtyard

dataset and our experimental setup, followed by the evaluation of our approach in Section 5.4. Finally, we conclude and summarize our work in Section 5.5.

5.2 Related Work

Early work in action recognition was mostly concerned with single actors in isolated scenes [4,5]. However, recently a lot of interest was directed towards modeling the complex interactions among observations explicitly. These interactions could be between scenes and actions [9], objects and actions [8, 10], or actions performed by two or more people [2, 11]. High-level and behavioral interactions were modeled using context-free grammars [7], AND-OR graphs [6, 47], dynamic Bayesian networks [13], network flow [39, 40], and probabilistic first-order logic [15, 16, 48]. However, one common assumption remained: action recognition was formulated as a multi-class problem. To the best of our knowledge, we are the first to formulate action recognition in a multi-label setting.

Recent work that uses attributes for action classification is conceptually related to our work. While attribute and multi-label classification share some of the techniques, semantically speaking they are very different problems. Liu *et al*. recognizes actions from videos by describing them with attributes (indoors, torso-twist, *etc*.) [49]. Yao *et al*. use a mixture of parts and attributes to classify actions in still images [50]. These attributes can represent a description of the action itself (indoors, two-handed), the pose needed (twisted torso, bent elbow, crossed legs), or a manipulated object part (bike seat, golf club). Both approaches classify multiple

binary attributes, whether in a pre-processing step or as latent variables, to eventually classify a single action performed by one person in the video or image. In contrast, we are concerned with busy scenes where actors can be performing multiple actions simultaneously, and we are interested in automatically understanding these actions and how they correlate. We accordingly represent the actions as a set of binary inter-dependent labels. Additionally, attributes can still be leveraged and have the potential to benefit multi-label action classification, but we leave this to future work.

Early approaches for multi-label classification reduced the problem to more common forms. McCallum proposed to view the problem as a multi-class classification problem with 2^L classes, representing the power set for L labels [51]. While extremely competitive, this approach is very computationally limiting. It also relies on the 0/1 loss and does not model the multi-label loss [1]. Boutell *et al.* also similarily proposed a power set classifier for multi-label scene classification [52], while Hsu *et al.* proposed a regression-based approach to map the label space to a lower dimensional vector space [53]. Elisseeff and Weston modeled the multi-label loss through a ranking solution [54], where more relevant labels are ranked higher than less relevant ones, and Cai and Hofmann used the same framework to model multi-label loss hierarchically on a tree [55].

Taskar proposed a max-margin structured prediction approach that can be applied to multi-label classification [56]. Structured prediction relies on inference during training, and generally exact inference in MRFs is intractable. Rousu *et al.* extended this to modeling hierarchical loss in a structured prediction setting using

a tree-structured model [57]. Restricting the model structure to a tree gives rise to many efficient inference approahes. More recently Petterson and Caetano leveraged MRFs with submodular pairwise interactions [58]. Submodularity also makes efficient inference possible through graph cut algorithms. Hariharan *et al.* took a middle approach by assuming a densely populated pairwise correlation matrix is fixed apriori [1]. Their approach generalizes one-vs-all classifiers in a principled way, and they propose efficient specialized optimization algorithms for it. While an apriori fixed correlation matrix can be expected to be given in a one-shot learning setting [59], it does not readily exist in a general multi-label setting. In our work we extend this approach and jointly optimize the multi-label training and discriminatively estimate the label correlations through a bilinear optimization problem, effectively learning the classifiers and the correlation matrix that together minimize the classification loss on the training set.

5.3 Approach

5.3.1 Formulation

We formulate multi-label classification in a max-margin framework. We are given N training samples and a set of L labels. Sample i is represented by $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \{\pm 1\}^L$, which are respectively its associated feature vector of dimensionality D and label vector of dimensionality L. Each label y_{il} is $+1$ if sample i is a positive sample for label l and -1 otherwise. We aim to optimize the classification loss function by jointly learning the classifier weight vectors and the label correlation

matrix. To this end, we optimize the following objective function [1]

$$F \equiv \quad \min_{\mathbf{W},\mathbf{P}} \frac{1}{2}\|\mathbf{W}\|_F^2 + \lambda\frac{1}{2}\|\mathbf{P}-\mathbf{I}_L\|_F^2 + C\sum_i \max_{\boldsymbol{y}} \left[\Delta(\mathbf{y}_i,\boldsymbol{y}) - (\mathbf{y}_i-\boldsymbol{y})^T\mathbf{P}^T\mathbf{W}^T\mathbf{x}_i\right], \tag{5.1}$$

where the bilinear classification function is represented by $\mathbf{y} = \mathbf{P}^T\mathbf{W}^T\mathbf{x}$. The hinge loss in Equation 5.1 penalizes the maximum margin violation for each sample under the loss function of interest. In our case, the loss function Δ represents the misclassification cost if one were to predict label $\boldsymbol{y}$ for $\mathbf{x}_i$ when the true label is $\mathbf{y}_i$.

Hariharan *et al*. introduced a special case of this formulation where they assumed that $\mathbf{P}$ was a known correlation matrix, apriori given or calculated [1]. Their resulting objective is only a function of $\mathbf{W}$. In contrast, we discriminatively learn $\mathbf{P}$ jointly with $\mathbf{W}$ so as to minimize the classification error on the training set. This, in turn, yields stronger bilinear classifiers but complicates the optimization. Our objective function is biconvex (as we will show), and we therefore approach it with an alternating optimization approach.

The formulation in Equation 5.1 has several advantages. A similar formulation that explicitly models the power set of labels, where the number of classes is 2^L, would equivalently require $N2^L$ constraints, regardless of the loss function used. This proves to be very limiting even for small values of L. On the other hand, Equation 5.1 under a decomposable loss function has only NL margin constraints. On a different

[1] Our hinge loss is defined similarly to the form commonly used in structured prediction [60, 61] and is therefore slightly different from that in [1].

note, modeling the dense pairwise correlations between the labels in a structured prediction framework renders inference, a required step in the optimization process, intractable. A common workaround is to restrict the graph to a tree structure or to impose constraints on the form of correlation (submodularity). In our case the label correlation matrix can be densely specified without negatively affecting the optimization problem.

5.3.2 Optimization

We approach the problem in Equation 5.1 using an alternating optimization algorithm. Given a fixed $\mathbf{P}$, we transform F to an SVM-like formulation by substituting $\mathbf{Z} = \mathbf{WP}$ and $\mathbf{R} = \mathbf{P}^T\mathbf{P} \succ 0$ (Positive Semi-Definite) to get the equivalent problem

$$\begin{aligned} G \equiv \quad & \min_{\mathbf{Z}} \frac{1}{2} tr(\mathbf{R}^{-1}\mathbf{Z}^T\mathbf{Z}) + \\ & C \sum_i \max_{\boldsymbol{y}} \left[\Delta(\mathbf{y}_i, \boldsymbol{y}) - (\mathbf{y}_i - \boldsymbol{y})^T \mathbf{Z}^T \mathbf{x}_i\right]. \end{aligned} \tag{5.2}$$

The regularization term for $\mathbf{P}$ becomes constant and is dropped. We next assume a decomposable loss function $\Delta(\mathbf{y}_i, \boldsymbol{y}) = \sum_l \delta_l(y_{il}, \boldsymbol{y})$, and then we set the loss function δ_l to the commonly used Hamming loss, inversely weighted by the class frequency for label l to account for class imbalance. For $y_{il} \in \{\pm 1\}$, this simplifies to $\delta_l(y_{il}, -y_{il})$ which we denote by δ_{il} for convenience. Putting everything together, we formulate the objective function equivalently in constrained form

$$
\begin{aligned}
G \equiv \quad & \min_{\mathbf{Z},\boldsymbol{\xi}} \frac{1}{2} tr(\mathbf{R}^{-1}\mathbf{Z}^T\mathbf{Z}) + C\sum_i \sum_l \xi_{il} \\
\text{s.t.} \quad & 2y_{il}\mathbf{z}_l^T\mathbf{x}_i \geq \delta_{il} - \xi_{il} \qquad \forall i, l \\
& \xi_{il} \geq 0 \qquad \forall i, l
\end{aligned} \tag{5.3}
$$

This is a quadratic matrix programming problem. It can be shown using a Schur complement argument that Equation 5.3 is convex in $\mathbf{Z}$ if and only if $\mathbf{R} \succ 0$, which is satisfied by definition.

An interesting case arises if we set $\mathbf{P} = \mathbf{I}_L$, where $\mathbf{I}_L$ is the identity matrix of size L. This corresponds to decorrelating the classifiers and recovers the following problem

$$
\begin{aligned}
G_0 \equiv \quad & \min_{\mathbf{Z},\boldsymbol{\xi}} \frac{1}{2} \|\mathbf{Z}\|_F^2 + C\sum_i \sum_l \xi_{il} \\
\text{s.t.} \quad & 2y_{il}\mathbf{z}_l^T\mathbf{x}_i \geq \delta_{il} - \xi_{il} \qquad \forall i, l \\
& \xi_{il} \geq 0 \qquad \forall i, l
\end{aligned} \tag{5.4}
$$

with in turn is equivalent to L completely independent linear classification subproblems

$$
\begin{aligned}
G_0 \equiv \quad & \sum_l S_l \\
\text{with } S_l \equiv \quad & \min_{\mathbf{z}_l, \boldsymbol{\xi}} \frac{1}{2}\mathbf{z}^T\mathbf{z} + C\sum_i \xi_i \\
\text{s.t.} \quad & 2y_{il}\mathbf{z}_l^T\mathbf{x}_i \geq \delta_{il} - \xi_i \qquad \forall i \\
& \xi_i \geq 0 \qquad \forall i
\end{aligned} \tag{5.5}
$$

This simple reduction motivated choosing the identity matrix as the regularization point for $\mathbf{P}$, *i.e.* the regularizer penalizes deviation from it. Similarly, in our optimization procedure, the initial value for $\mathbf{P}$ is $\mathbf{I}_L$. Additionally, this turned out to be an appropriate baseline in our experiments, corresponding to 1-vs-all linear SVM classifiers for the action labels, which is a commonly used benchmark for multi-label methods [52, 62].

We further reduce the number of constraints by employing a one-slack formulation instead [60]. The idea is to replace the N constraints on the hinge loss, one for each of the training samples, with a single constraint on the sum of the hinge losses for all the samples, hence we replace ξ_{il} with one slack variable per label ξ_l. It can be shown that the solution to the one-slack formulation is extremely sparse and is equivalent to the solution to the original problem if $\xi_l^* = \frac{1}{N}\sum_i \xi_{il}^*$, where $\boldsymbol{\xi}^*$ is the slack vector at the minimum solution [60].

We proceed to solve the one-slack formulation of Equation 5.3 using a cutting plane approach [61]. At each iteration we find the violated constraints for all the training samples, and we append them to the working set. This algorithm terminates

Algorithm 1 Cutting plane algorithm for $\mathbf{P}$

INPUT: $\mathbf{V}, \mathbf{Y}, C, \epsilon$

$\mathcal{W} = \emptyset$

repeat

$\mathcal{P} = \{\mathbf{P} : (p_{ij} = p_{ji} \wedge p_{ij} \geq -1 \wedge p_{ij} \leq 1) \; \forall \; i, j \; \wedge$

$\frac{2}{N}\mathbf{p}_l^T \sum_i c_{il} y_{il} \mathbf{v}_i \geq \frac{1}{N} \sum_i c_{il} \delta_{il} - \zeta_l \; \forall \; \boldsymbol{c} \in \mathcal{W}\}$

$\{\mathbf{P}, \boldsymbol{\zeta}\} = \underset{\mathbf{P} \in \mathcal{P}, \boldsymbol{\zeta} > 0}{\arg\min} \lambda \frac{1}{2} \|\mathbf{P} - \mathbf{I}_L\|_F^2 + C \sum_l \zeta_l$

for $l = 1 \ldots L$ **do**

$$c_{il} = \begin{cases} 1 & 2 y_{il} \mathbf{p}_l^T \mathbf{v}_i \leq \delta_{il} \\ & \forall i \\ 0 & otherwise \end{cases}$$

$\mathcal{W} = \mathcal{W} \cup \{\boldsymbol{c}\}$

until $\max_l (\frac{1}{N} \sum_i c_{il} \delta_{il} - \frac{2}{N} \mathbf{p}_l^T \sum_i c_{il} y_{il} \mathbf{v}_i - \zeta_l) \leq \epsilon$

OUTPUT: P

in a number of iterations independent of the output space size [61], and in our experiments we needed fewer than 50 iterations to converge and were faster than the implementation from [1]. The process is detailed in Algorithm 1.

Solving Equation 5.3 we find $\mathbf{Z}$, and we can then recover $\mathbf{W} = \mathbf{Z}\mathbf{P}^{-1}$. Similarly, given a fixed $\mathbf{W}$, we can turn F to an SVM-like formulation by first transforming the feature vectors to $\mathbf{v}_i = \mathbf{W}^T \mathbf{x}_i$, where each $\mathbf{v}_i$ is of size L, to get the equivalent problem

Algorithm 2 Cutting plane algorithm for $\mathbf{Z}$

1: **INPUT:** $\mathbf{X}, \mathbf{Y}, \lambda, C, \epsilon$

2: $\mathcal{W} = \emptyset$

3: **repeat**

4: $\quad \mathcal{Z} = \{\mathbf{Z} : \frac{2}{N}\mathbf{z}_l^T \sum_i c_{il} y_{il} \mathbf{x}_i \geq \frac{1}{N} \sum_i c_{il} \delta_{il} - \xi_l \ \forall \ \boldsymbol{c} \in \mathcal{W}\}$

5: $\quad \{\mathbf{Z}, \boldsymbol{\xi}\} = \underset{\mathbf{Z} \in \mathcal{Z}, \boldsymbol{\xi} > 0}{\arg\min} \frac{1}{2} tr(\mathbf{R}^{-1}\mathbf{Z}^T\mathbf{Z}) + C \sum_l \xi_l$

6: $\quad$ **for** $l = 1 \ldots L$ **do**

7: $\quad\quad c_{il} = \begin{cases} 1 & 2y_{il}\mathbf{z}_l^T\mathbf{x}_i \leq \delta_{il} \\ 0 & otherwise \end{cases} \quad \forall i$

8: $\quad \mathcal{W} = \mathcal{W} \cup \{\boldsymbol{c}\}$

9: **until** $\max_l (\frac{1}{N} \sum_i c_{il}\delta_{il} - \frac{2}{N}\mathbf{z}_l^T \sum_i c_{il} y_{il} \mathbf{x}_i - \zeta_l) \leq \epsilon$

10: **OUTPUT: Z**

$$H \equiv \quad \min_{\mathbf{P}} \lambda \frac{1}{2} \|\mathbf{P} - \mathbf{I}_L\|_F^2 + C \sum_i \max_{\boldsymbol{y}} \left[\Delta(\mathbf{y}_i, \boldsymbol{y}) - (\mathbf{y}_i - \boldsymbol{y})^T \mathbf{P}^T \mathbf{v}_i\right]. \tag{5.6}$$

Under the same decomposable loss function Δ previously introduced, we reformulate the objective function equivalently in constrained form

$$
\begin{aligned}
H \equiv \quad & \min_{\mathbf{P},\boldsymbol{\zeta}} \lambda \frac{1}{2} \|\mathbf{P} - \mathbf{I}_L\|_F^2 + C \sum_i \sum_l \zeta_{il} \\
\text{s.t.} \quad & 2 y_{il} \mathbf{p}_l^T \mathbf{v}_i \geq \delta_{il} - \zeta_{il} \qquad \forall i, l \\
& \zeta_{il} \geq 0 \qquad \forall i, l
\end{aligned} \tag{5.7}
$$

Equation 5.7 is a convex quadratic programming problem. To enforce $\mathbf{P}$ to be a symmetric correlation matrix, we add the constraints $p_{ij} = p_{ji}$, $p_{ij} \geq -1$, and $p_{ij} \leq 1$. We then transform the problem to a one-slack formulation as before, replacing ζ_{il} with one slack variable per label ζ_l. The resulting optimization problem is also solved using a cutting plane algorithm, where we iteratively find the violated constraints for all the training samples, and append them to the working set. The process is detailed in Algorithm 2.

Our alternating optimization approach is illustrated in Algorithm 3. We start by initializing $\mathbf{P}$ to $\mathbf{I}_L$. We then proceed to alternate between fixing $\mathbf{P}$ and solving for $\mathbf{W}$, and then fixing $\mathbf{W}$ and solving for $\mathbf{P}$.

5.4 Experiments

5.4.1 Setup

Given that there are no multi-label action recognition datasets, we set out to relabel an existing datasets for our task. Datasets like KTH [5] and Weizmann [4] feature only a single actor in isolated scenes and are therefore not suitable for a multi-label setting. Similarly, the UT Interaction dataset [63] only features a single

Algorithm 3 Learning Bilinear Multi-Label Classifiers

INPUT: $\mathbf{X}, \mathbf{Y}, \lambda, C, \epsilon, T$

for $t = 1 \dots T$ **do**

  if $t = 1$ **then**

    Set $\mathbf{P}_t = \mathbf{I}_L$

  else

    Set $\mathbf{v}_i = \mathbf{W}_{t-1}^T \mathbf{x}_i \quad \forall i$

    Calculate $\mathbf{P}_t$ from Algorithm 1

  Set $\mathbf{R} = \mathbf{P}_t^T \mathbf{P}_t$

  Calculate $\mathbf{Z}_t$ from Algorithm 2

  Set $\mathbf{W}_t = \mathbf{Z}_t \mathbf{P}_t^{-1}$

  if $\max |\mathbf{Z}_t - \mathbf{Z}_{t-1}| < \epsilon$ **then**

    break

OUTPUT: $\mathbf{P}_t$ and $\mathbf{W}_t$

action between two actors. On the other hand, we considered the Collective Activity dataset [2]. The dataset features multiple people in different situations, but in most videos all the actors were performing the same action (*e.g.*, dancing), which unfortunately also made it unsuitable for our task.

We set out to relabel the UCLA Courtyard dataset, which features two different bird's eye viewpoints of the same courtyard at the UCLA campus [47]. The dataset features six high resolution videos of many actors in a natural setting performing a variety of actions on both the individual level and the group level. Each

actor is annotated by one of 8 orientations, one of 7 poses, and one of 10 individual actions: 1. riding a skateboard, 2. riding a bike, 3. riding a scooter, 4. driving a car, 5. walking, 6. talking, 7. waiting, 8. reading, 9. eating, and 10. sitting. We used the same set of labels for our multi-label experiments. The dataset was evenly split (50-50%) for training and testing, maintaining similar class label distributions for the two halves [47].

Similar to Amer *et al*. [47], we extracted and normalized Histogram of Oriented Gradients (HOG) [22] features around motion-based STIP features and Histogram of Optical Flows (HOF) [64] around KLT tracks from the bounding box of each actor, and therefore the spatial and temporal characteristics were implicitly accounted for through the feature descriptors.

Ultimately the dataset contains over 4.4 million frames, and therefore manually relabeling the entire dataset is very time-consuming. We resorted to bootstrapping the relabeling process: using the current annotations (pose, orientation, individual action, group action, group orientation, *etc*.), we predict a new set of action labels that include the current action label among others. For instance, a person labeled as *eating* while facing another person, both part of a group labeled as *sitting*, is relabeled as *sitting*, *eating*, and *talking*. We first ran the labels through a large set of similar relabeling rules and then we manually inspected the outcome and optimized the rules to correct any erroneous labels as necessary. This process was repeated a few times to ensure high fidelity for the groundtruth labels. Figure 5.2 shows a sample frame with multi-label actions. Given the high resolution of the dataset, we zoomed in on a few groups. While the labels for the top group did not

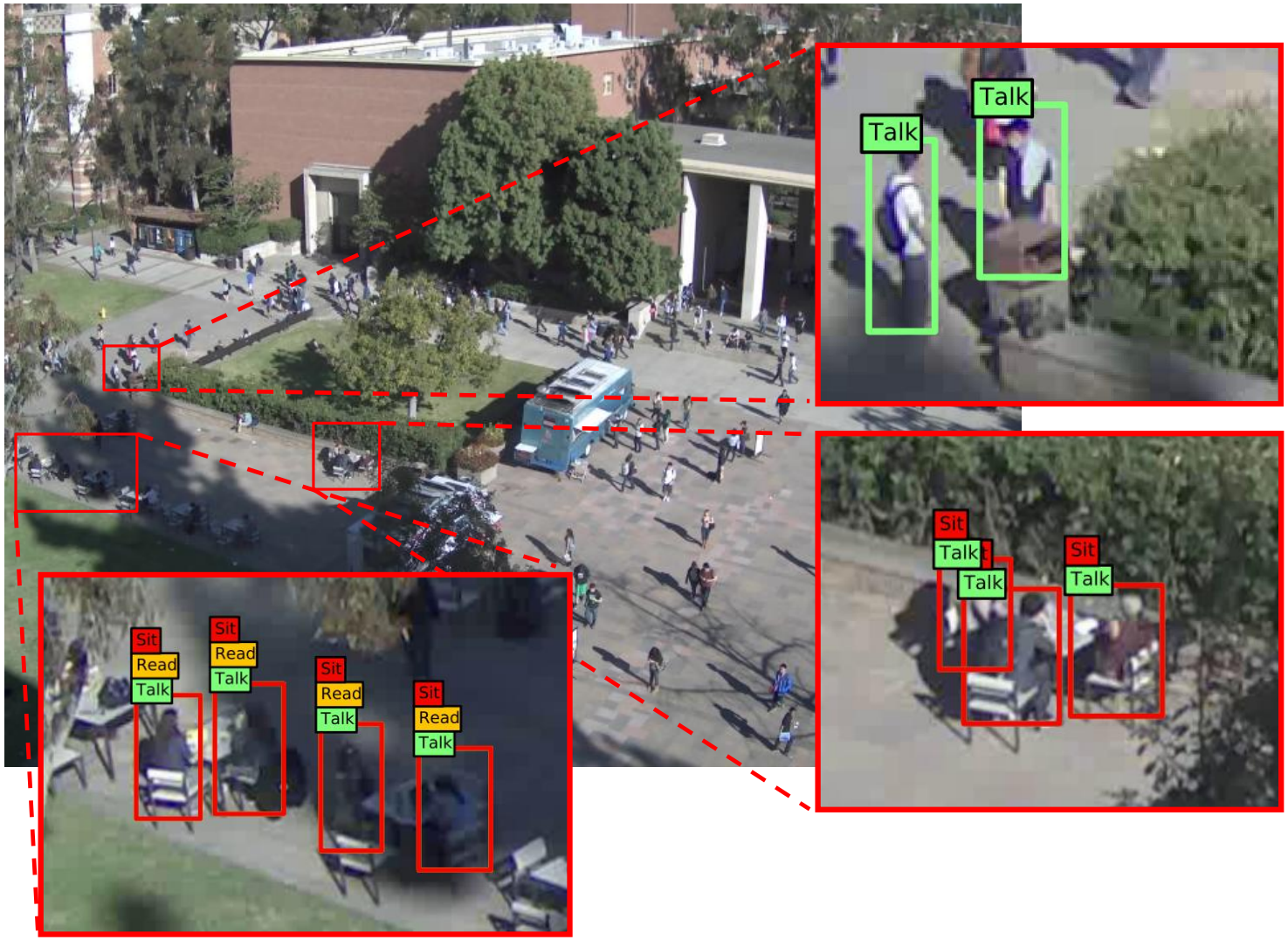

Figure 5.2: A sample frame from the relabeled UCLA Courtyard dataset.In the resulting labels, 56.9% of all actors are performing two or more actions at the same time and 4.9% are performing three or more actions.

change, other groups received additional appropriate action labels. Relabeling was bootstrapped using rules that took into account all the dataset annotations (pose, orientation, individual action, group action, group orientation, *etc.*) to predict new action labels. In the resulting labels, 56.9% of all actors are performing two or more actions at the same time and 4.9% are performing three or more actions.

5.4.2 Results

Since we initialize the label correlation matrix in our algorithm to the identity matrix $\mathbf{I}_D$, the binary classifiers trained after the first iteration correspond to 1-vs-all linear SVMs trained independently on the same features. This is equivalent to disregarding label correlations and just optimizing Equation 5.5. Independently training label classifiers in a multi-label setting is an appropriate standard baseline [52, 62], which in our algorithm corresponds to the output after the first iteration. This allows us to evaluate the performance improvement through the iterations by the optimization algorithm. Additionally, we implemented the multi-label approach of Hariharan *et al.* [1] as a second baseline, where the label correlation matrix is estimated offline from the training data as: $\frac{1}{N}\sum_{i}^{N}\mathbf{y}_i\mathbf{y}_i^T$. Our experiments verify that our approach that discriminatively learns the correlations yields better classification performance.

We report our quantitative results in Table 5.1. While we are using similar features and data splits to Amer *et al.* [47], we are learning with an entirely different label set, and therefore we cannot directly compare to their results. We include the

numbers nonetheless due to the lack of multi-label action recognition datasets and benchmarks. We report the per-class accuracies as well as the mean over all classes. We also report the Hamming loss over all testing samples, which is a common measure for multi-label classification.

As can be seen from the table, our baseline classifier performance is very competitive. We attribute the significant improvement in the mean accuracies to using the weighted hamming loss, in contrast to the Hamming loss (0-1) which optimizes the total classification accuracy. The per-label accuracies for classes like *driving a car*, which looks very unique compared to other classes, is already at 100% after the first iteration. The algorithm converged after 5 iterations of alternating optimization. The improvements, on average, are consistent through the iterations, and more specifically, labels like *reading* and *sitting* received the highest gain through the label correlations, presumably through the correlation with labels like *eating*. Similarly the accuracy for *riding a scooter* also significantly increased, presumably through the correlation with *sitting*. The Hamming loss also decreased through the optimization. It did however slightly increase the last iteration, which again can be attributed to using the weighted hamming loss, which further increased the mean accuracy but slightly sacrificed the total accuracy (1 - Hamming loss).

We also visualize the final label correlation matrix $\mathbf{P}$ calculated by our algorithm in Figure 5.3. Lighter shades, as seen on the main diagonal, denote positive correlations, and darker shades denote negative (or inverse) correlations. Some of the learned correlations are very intuitive. For example, *walking* and *talking* are likely to co-occur at the same time, which is accurately reflected in the matrix. In

Approach	Walk	Wait	Talk	Car	Skate	Scooter	Bike	Read	Eat	Sit	Avg	Loss
1-vs-all	72.9	69.7	64.7	100.0	50.2	71.4	51.5	66.2	100.0	83.0	73.0	11.8
[1]	68.8	73.9	65.3	100.0	54.9	73.1	54.9	76.1	94.6	87.6	74.9	9.8
Our model	70.6	74.7	68.6	100.0	56.8	91.7	58.3	95.2	100.0	87.4	80.3	9.0

Table 5.1: The quantitative results of our approach. Our joint approach to the problem improves on the 1-vs-all baseline as well as the two-stage approach of Hariharan *et al.* [1].

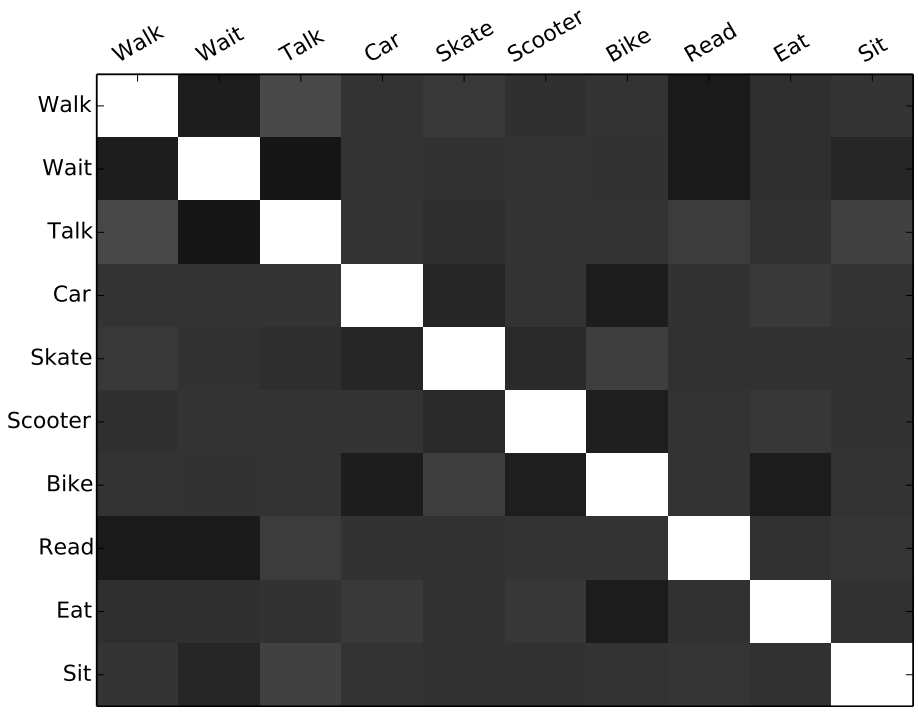

Figure 5.3: A visualization of the final label correlation matrix $\mathbf{P}$. Intuitively, *walking* and *talking* are positively correlated, while *walking* and *waiting* were unlikely to co-occur in the dataset.

contrast, *eating* and *biking* are inversely correlated as expected.

5.5 Conclusion

We posed action recognition as a multi-label classification problem. Instead of limiting each actor in a natural scene to a single label, we proposed a multi-label setting that is more natural to the problem. Multi-label classification has been either reduced to more common forms, such as multi-class classification, or treated as a Markov Random Field labeling in a structured prediction setting, but both ap-

proaches suffer from drawbacks. We instead extended recent work on max-margin multi-label classification to the case where the label correlation matrix is not apriori known, and we posed the multi-label classification and label correlation estimation as a joint problem. We then devised an alternating optimization algorithm to minimize the coupled problem. Finally, given the lack of multi-label action recognition datasets, we relabeled the UCLA Courtyard dataset for our task. We report state-of-the-art results on the dataset using our approach. In future work we plan to investigate integrating group activities into our framework.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

In this thesis we proposed models that integrate different useful cues to aid activity recognition. While activity recognition has been typically approached in isolation, we identify spatiotemporal and contextual cues that we model as latent variables in a joint formulation. By optimizing the objective function for this formulation, we can recover more accurate activity labels and simultaneously additional track-level and scene-level information. We tackled this problem through a mathematical optimization approach based on dual decomposition, as well as a probabilistic soft logic approach. Additionally, we proposed a model to cast action recognition as a multi-label problem, where we jointly learn the pairwise action label correlations.

While our work frees action recognition from some of the preprocessing steps necessary in many pipelines (*e.g.* tracking), the proposed models still require bounding boxes to be detected beforehand. This contributes to a solution that depends on multiple stages of processing, where errors in a person detection subsystem, through either false positives or false negatives, would propagate and negatively affect the entire pipeline.

Additionally, the proposed models were not adapted to an online setting. All the frames of an entire video have to go through the pipeline at the same time, which is a major limitation. A direct solution in this case would be to apply the models on time windows, and propagate the solution forward. This might be an adequate online adaptation in many cases, at the expense of a reduction in accuracy.

5.2 Future Work

The limitations mentioned in the previous section open the door for interesting future extensions. Some of these extensions are very open-ended and will likely require new datasets for a rigorous evaluation. A list of possible directions include:

- Adapting the joint models to an online setting. A trivial solution would be to apply the model to time windows, propagating the solution forward. However, a recursive formulation, for instance, that explicitly models the uncertainty in the propagated solution from previous frames might perform better.

- Jointly modeling person detection, action recognition, and tracking. While our work jointly models action recognition and tracking, recently Wu *et al.* jointly modeled detection and tracking [65]. A system that integrates all three problems in a single step is likely to outperform our approach, as well as eliminate another stage in the pipeline.

- Explicitly modeling groups of people. Choi *et al.* looked at group discovery very recently [66] by modeling inter- and intra-group interaction patterns. Explicitly integrating this problem in an action recognition system, where action

classifiers can benefit from group labels and vice versa, is a possible research topic. This can also be a viable extension to the multi-label formulation.

- Utilizing additional cues for action recognition. This includes jointly modeling object detection or recognition as a cue for action recognition, where for instance holding a shovel is a good indicator that this person is digging. The same goes for jointly modeling semantic segmentation and action recognition, where standing on a crosswalk is a strong cue that the person is crossing the street.

- Building large-scale approaches that jointly model the same cues. Deep learning models are a class of models that can leverage and continue to improve with large amounts of data [67, 68], and they were recently shown to significantly outperform many other approaches on a variety of recognition tasks [69, 70]. Training a deep model that integrates all these cues into a large-scale action recognition system would be an interesting direction to investigate.

Bibliography

[1] Bharath Hariharan, Lihi Zelnik-Manor, S. V. N. Vishwanathan, and Manik Varma. Large scale max-margin multi-label classification with priors. In *International Conference on Machine Learning*, 2010.

[2] W. Choi, K. Shahid, and S. Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *International Workshop on Visual Surveillance*, 2009.

[3] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Computer Vision and Pattern Recognition*, 2011.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *International Conference on Computer Vision*, 2005.

[5] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition*, 2004.

[6] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *Computer Vision and Pattern Recognition*, 2009.

[7] M. S. Ryoo and J. K. Aggarwal. Stochastic representation and recognition of high-level group activities. *International Journal of Computer Vision*, 93(2):183–200, 2010.

[8] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. *Computer Vision and Pattern Recognition*, 2010.

[9] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Computer Vision and Pattern Recognition*, 2009.

[10] Abhinav Gupta and Larry S. Davis. Objects in action: An approach for combining action understanding and object perception. In *Computer Vision and Pattern Recognition*, 2007.

[11] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *Neural Information Processing Systems*, 2010.

[12] T. Lan, Y. Wang, G. Mori, and S. N. Robinovitch. Retrieving actions in group contexts. In *ECCV Workshop on Sign, Gesture, and Activity*, 2010.

[13] Tao Xiang and Shaogang Gong. Beyond tracking: modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67:21–51, 2006.

[14] Asaad Hakeem and Mubarak Shah. Learning, detection and representation of multi-agent events in videos. *Artificial Intelligence*, 2007.

[15] Vlad I. Morariu and Larry S. Davis. Multi-agent event recognition in structured scenarios. In *Computer Vision and Pattern Recognition*, 2011.

[16] William Brendel, Sinisa Todorovic, and Alan Fern. Probabilistic event logic for interval-based event recognition. In *Computer Vision and Pattern Recognition*, 2011.

[17] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition*, 2008.

[18] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition*, 2008.

[19] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition*, 2011.

[20] J. Berclaz, F. Fleuret, E. Tretken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.

[21] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *International Conference on Computer Vision*, 2011.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005.

[23] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[24] C. Sutton and A. McCallum. Piecewise training for undirected models. In *Uncertainty in Artificial Intelligence*, 2005.

[25] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006.

[26] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[27] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum-weight independent set. In *Computer Vision and Pattern Recognition*, 2011.

[28] K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *International Conference on Machine Learning*, 2008.

[29] Sameh Khamis, Vlad I. Morariu, and Larry S. Davis. A flow model for joint action recognition and identity maintenance. In *Computer Vision and Pattern Recognition*, 2012.

[30] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[31] N. Komodakis, N. Paragios, and G. Tziritas. Mrf optimization via dual decomposition: Message-passing revisited. In *International Conference on Computer Vision*, 2007.

[32] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI Conference on Artificial Intelligence*, pages 133–136, 1982.

[33] David Gamarnik, Devavrat Shah, and Yehua Wei. Belief propagation for min-cost network flow: convergence & correctness. In *ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[34] J. M. Gonfaus, X. Boix, J. Van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez. Harmony potentials for joint classification and segmentation. In *Computer Vision and Pattern Recognition*, 2010.

[35] S. Bach, M. Broecheler, L. Getoor, and D. O'Leary. Scaling MPE inference for constrained continuous markov random fields with consensus optimization. In *Neural Information Processing Systems*, 2012.

[36] Stephen H. Bach, Bert Huang, Ben London, and Lise Getoor. Hinge-loss markov random fields: Convex inference for structured prediction. In *Uncertainty in Artificial Intelligence*, 2013.

[37] M. Broecheler, L. Mihalkova, and L. Getoor. Probabilistic similarity logic. In *Uncertainty in Artificial Intelligence*, 2010.

[38] A. Kimmig, S. Bach, M. Broecheler, B. Huang, and L. Getoor. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012.

[39] Sameh Khamis, Vlad I. Morariu, and Larry S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *European Conference on Computer Vision*, 2012.

[40] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*, 2012.

[41] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[42] J. Neville and D. Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8:653–692, 2007.

[43] B. Taskar, M. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing Systems*, 2003.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 2011.

[45] M. Collins. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Empirical Methods in Natural Language Processing*, 2002.

[46] D. Lowd and P. Domingos. Efficient weight learning for Markov logic networks. In *Principles and Practice of Knowledge Discovery in Databases*, 2007.

[47] Mohamed R. Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *European Conference on Computer Vision*, 2012.

[48] Ben London, Sameh Khamis, Stephen H. Bach, Bert Huang, Lise Getoor, and Larry S. Davis. Collective activity detection using hinge-loss markov random fields. In *CVPR Workshop*, 2013.

[49] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition*, 2011.

[50] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas J. Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision*, 2011.

[51] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop*, 1999.

[52] M. Boutell, J. Luo, X. Shen, and C. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):17571771, 2004.

[53] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *Neural Information Processing Systems*, 2009.

[54] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Neural Information Processing Systems*, 2001.

[55] L. Cai and T. Hofmann. Exploiting known taxonomies in learning overlapping concepts. In *International Joint Conference on Artificial Intelligence*, 2007.

[56] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Neural Information Processing Systems*, 2003.

[57] J. Rousu, C. Saunders, S. Szedmak, and J. ShaweTaylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:16011626, 2006.

[58] J. Petterson and T. S. Caetano. Submodular multi-label learning. In *Neural Information Processing Systems*, 2011.

[59] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition*, 2009.

[60] T. Joachims. Training linear SVMs in linear time. In *ACM SIGKDD*, 2006.

[61] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:14531484, 2005.

[62] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.

[63] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, International Conference on Pattern Recognition contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[64] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition*, 2008.

[65] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *Computer Vision and Pattern Recognition*, 2012.

[66] W. Choi, Y. Chao, C. Pantofaru, and S. Savarese. Discovering groups of people in images. In *European Conference on Computer Vision*, 2014.

[67] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[68] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *IEEE*, 1998.

[69] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *International Conference on Machine Learning*, 2011.

[70] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 2012.